INDUSTRIAL CONTROL COMMUNICATIONS, INC.

Instruction Manual

EIP4CCPU

February 4, 2011
ICC #10816

© 2011 Industrial Control Communications, Inc.

EIP4CCPU
User's Manual

Part Number 10816
Printed in U.S.A.
©2011 Industrial Control Communications, Inc.
All rights reserved

NOTICE TO USERS

Industrial Control Communications, Inc. reserves the right to make changes and improvements to its products without providing notice.

Industrial Control Communications, Inc. shall not be liable for technical or editorial omissions or mistakes in this manual, nor shall it be liable for incidental or consequential damages resulting from the use of information contained in this manual.

INDUSTRIAL CONTROL COMMUNICATIONS, INC.'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE-SUPPORT DEVICES OR SYSTEMS. Life-support devices or systems are devices or systems intended to sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling and user's manual, can be reasonably expected to result in significant injury.

No complex software or hardware system is perfect. Bugs may always be present in a system of any size. In order to prevent danger to life or property, it is the responsibility of the system designer to incorporate redundant protective mechanisms appropriate to the risk involved.

This user's manual may not cover all of the variations of interface applications, nor may it provide information on every possible contingency concerning installation, programming, operation, or maintenance.

The contents of this user's manual shall not become a part of or modify any prior agreement, commitment, or relationship between the customer and Industrial Control Communications, Inc. The sales contract contains the entire obligation of Industrial Control Communications, Inc. The warranty contained in the contract between the parties is the sole warranty of Industrial Control Communications, Inc., and any statements contained herein do not create new warranties or modify the existing warranty.

Any electrical or mechanical modifications to this equipment without prior written consent of Industrial Control Communications, Inc. will void all warranties and may void any UL/cUL listing or other safety certifications. Unauthorized modifications may also result in equipment damage or personal injury.

APPLICABLE VERSIONS

EIP4CCPU Firmware V1.100
Q12DCCPU-V Serial Number 12082XXXXXXXXX or later

TABLE OF CONTENTS

1. Introduction 5

2. Features 6

3. Scanner Concepts 7

4. Installation 8

4.1 License File 8
4.2 Preparing Q12DCCPU-V 9
4.3 Preparing CompactFlash Card 9
4.4 Power-On 9

5. LED Indicators 10

5.1 RUN 10
5.2 CF CARD 10
5.3 ERR. 10
5.4 7-Segment Display 10
5.5 Ethernet Jack 10

6. DIP Switches 11

6.1 RUN/STOP/MODE 11
6.2 RES/SEL 11

7. Event Codes 12

8. Software Specifications 16

8.1 Overall 16
8.2 Client Class 1 Implicit (I/O) Messaging 16
8.3 Client Explicit Messaging 16
8.4 Server Class 3 Explicit Messaging 17

9. Protocol-Specific Information 19

9.1 EtherNet/IP 19
9.1.1 Overview 19
9.1.2 EtherNet/IP Client 19
9.1.3 EtherNet/IP Server 24
9.2 Allen Bradley CSP 24
9.2.1 Overview 24
9.2.2 CSP Client 24
9.2.3 CSP Server 24
9.3 CIP Status Codes 25

10. EIP4CCPU Scanner Configuration Utility 36

10.1 Features 36
10.2 Installation 36
10.3 Firewall Settings 39
10.4 User Interface 42
10.5 Register EDS Files 43
10.6 Unregister EDS Files 45
10.7 Device Discovery Wizard 46
10.8 Discover Local Scanners 48
10.9 Adding a Scanner 48

10.10 Default Configuration 49
10.11 Configuring the Scanner 49
10.11.1 General 49
10.11.2 CPU Settings 50
10.11.3 Explicit Messaging 51
10.12 Adding a Target Device and Configuring Connection Properties 52
10.13 Delete or Edit Connections 57
10.14 Delete Device 57
10.15 Downloading the Configuration 57
10.16 Uploading the Configuration 58
10.17 Exporting the Scanner's Configuration 58
10.18 Importing the Scanner's Configuration 58
10.19 Saving the Project 59

11. How to Configure Modular I/O 60

12. RSNetWorx Example 64

13. FAQ 72

1. Introduction

Congratulations on your purchase of EIP4CCPU EtherNet/IP scanner firmware. This firmware runs on the Q12DCCPU-V and effectively transforms the Q12DCCPU-V into an EtherNet/IP scanner. The term “scanner” shall be used to reference the EIP4CCPU firmware running on the Q12DCCPU-V. The scanner allows communication to EtherNet/IP devices and is configured over Ethernet.

Before using the scanner, please familiarize yourself with the product and be sure to thoroughly read the instructions and precautions contained in this manual. In addition, please make sure that this instruction manual is delivered to the end user of the scanner, and keep this instruction manual in a safe place for future reference.

For the latest information, support software and firmware releases, please visit http://www.iccdesigns.com.

Note that different firmware versions may provide varying levels of support. When using this manual, therefore, always keep in mind that the version indicated on your unit must be listed on page 2 for all documented aspects to apply.

This manual will primarily be concerned with the scanner’s software specifications, installation, configuration, and operational characteristics.

To maximize the abilities of the scanner, a working familiarity with this manual will be required. This manual has been prepared for the scanner installer, user, and maintenance personnel. With this in mind, use this manual to develop a system familiarity before attempting to install or operate the scanner.

Please note that the hardware system must support the multiple CPU high speed transmission (high speed shared memory). This manual assumes that the user is familiar with:

- CIP specification (can be obtained from ODVA)
- Multiple CPU system configuration
- High speed shared memory
- GX Developer or GX Works2
- CompactFlash card and CompactFlash card reader
- FTP (File transfer protocol)

2. Features

Supported Protocols

The scanner currently provides support for the following Ethernet protocols:

- EtherNet/IP Client (Scanner)
  - Class 1 I/O
  - Class 3 Explicit Messaging
- EtherNet/IP Server (Adapter)
  - Class 3 Explicit Messaging
- CSP (also known as PCCC or Allen Bradley Ethernet) Client
- CSP Server

Device Discovery Wizard

The EIP4CCPU Scanner Configuration Utility can discover and configure the network settings of a scanner regardless of whether or not its network parameters are compatible with the subnet upon which it resides.

Dynamic Configuration

Configurations can be downloaded to the scanner and take effect immediately without reboot (using RSNetWorx).

Supports Modular I/O

Supports devices such as FLEX IO and POINT IO.

Field-Upgradeable

As new firmware becomes available, the scanner can be upgraded in the field by the end-user. Simply overwrite the current firmware file and license file (if applicable) located on the root directory of the CF card with the new firmware file and license file (if applicable) and reboot. Please contact your local distributor to determine if an upgrade fee may apply.

3. Scanner Concepts

The scanner is configured using the EIP4CCPU Scanner Configuration Utility (ESCU) or RSNetWorx. The scanner communicates with EtherNet/IP devices using the EtherNet/IP communication protocol and supports initiation of both class 1 I/O and class 3 explicit messaging connections. It additionally supports the class 3 explicit messaging server. Only Ethernet port 1 (CH1) is used by the scanner.

The scanner is paired with a sequencing PLC. Therefore, the scanner does not host its own database; rather, data is exchanged with the sequencing PLC. For class 1 connections, I/O data is exchanged via the high speed shared memory. The user must configure the shared memory settings for the system, which will determine the maximum amount of I/O data that can be processed by the scanner.

The other fundamental aspect of the scanner is the concept of a configurable “explicit message instruction”. An “explicit message instruction” is used to describe what service (read or write) is to be requested on the network using a class 3 connection. The scanner will service the “explicit message instruction” when the instruction is triggered by the program executing on the sequencing PLC.

Data values are implicitly assumed to be 16-bit unsigned values, unless otherwise noted, and are not modified by the scanner. It is the responsibility of the end user to apply any necessary data conversions (scaling, floating point conversion, offset, etc).

4. Installation

4.1 License File

To obtain a license file, email ICC at sales@iccdesigns.com and include the entitlement key and the serial number of the Q12DCCPU-V. Entitlement keys are purchased from ICC. The serial number of the Q12DCCPU-V is located on the front (at the bottom) of the module (see Figure 1) and also on the side face (see Figure 2). A license file is required for unlimited run-time. For evaluation purposes, the scanner will operate in trial mode if no license file is present. Full functionality of the scanner is maintained in trial mode for a period of one hour. Once the trial mode expires, all EtherNet/IP activity will terminate until the scanner is rebooted.

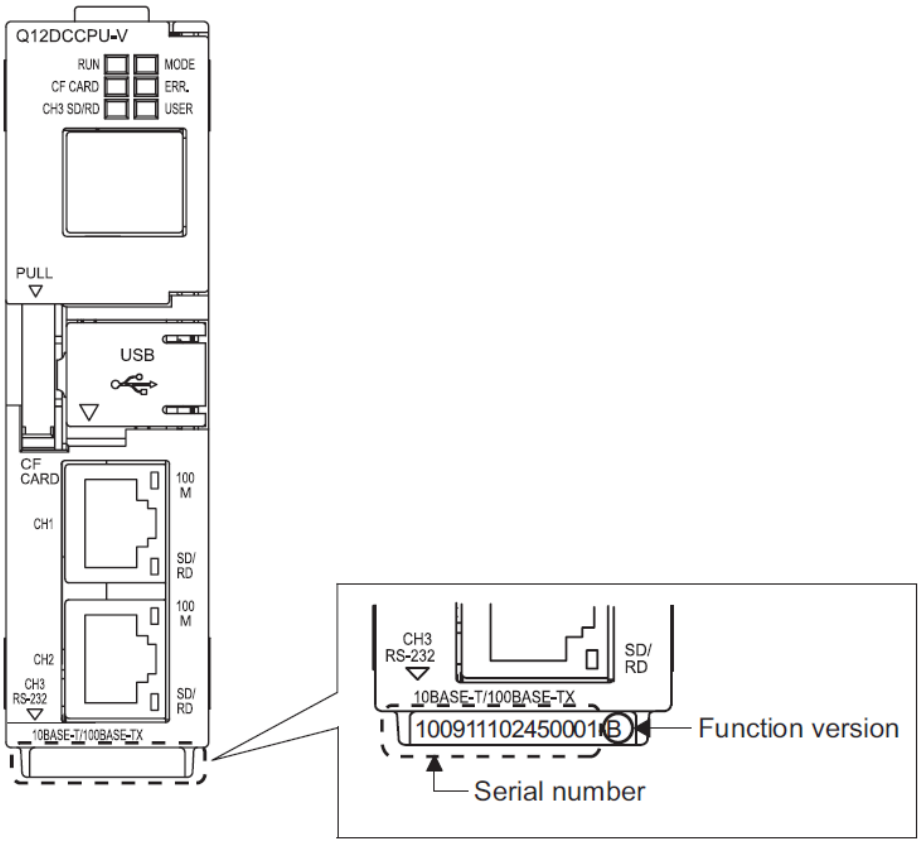


Figure 1: Serial Number Front

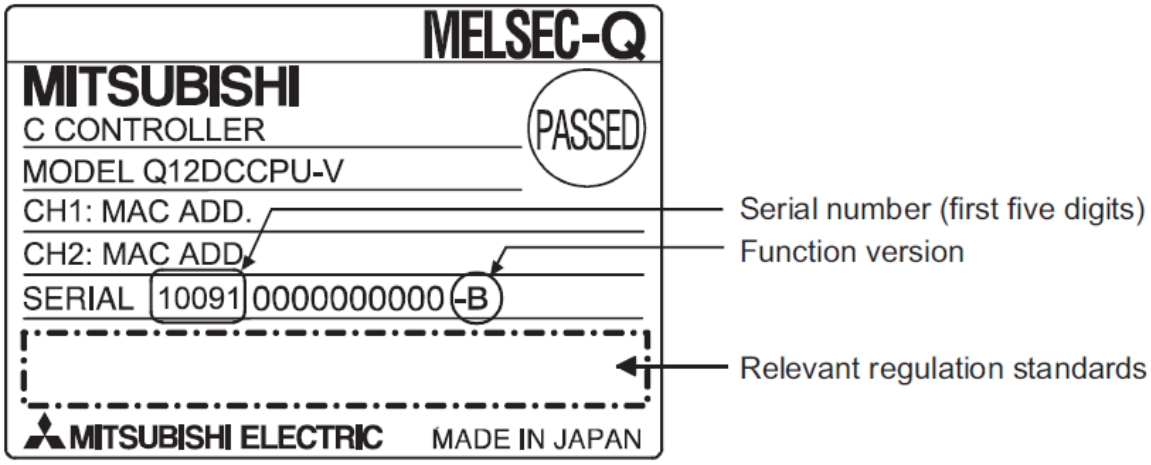


Figure 2: Serial Number Side Face

4.2 Preparing Q12DCCPU-V

Ensure that a sequencing PLC resides on the high speed bus base unit. Insert the Q12DCCPU-V module into the base unit at the next available slot. The Q12DCCPU-V must reside to the right of any sequencing PLCs. There can be at most four controllers, including the Q12DCCPU-V, on the base unit. Set the RUN/STOP/MODE switch on the Q12DCCPU-V to RUN.

4.3 Preparing CompactFlash Card

Preparation of the CompactFlash (CF) card is not necessary if one was already provided by the seller. Otherwise complete the following steps:

1. Ensure the CF card is compliant with the Q12DCCPU-V.
2. If the CF card has not been formatted, please format the CF card at this time. A CF card with a capacity of 64MB is more than sufficient.
3. Locate and copy the files “EIP4CCPU.OUT”, “STARTUP.CMD”, and the appropriate license file to the root directory of the CF card.
4. If the EIP4CCPU firmware was purchased via a distributor, please contact that distributor for any additional files. Additional files must also be copied to the root directory of the CF card.
5. Insert the CF card into the Q12DCCPU-V.

4.4 Power-On

Power on the CPUs. If the ERR LEDs are blinking, there may be an issue with the system configuration or multiple CPU settings. Upon powering up, the scanner will attempt to synchronize itself with the multiple CPU settings of the sequencing PLC. If event code 50 is flashing on the 7-segment LED display, power-cycle the CPUs and verify that the ERR LEDs are no longer blinking.

5. LED Indicators

The Q12DCCPU-V contains several different LED indicators, each of which conveys important information about the status of the unit. Only the LEDs specific to scanner functionality are summarized here.

5.1 RUN

The RUN LED must be solid green. If not, check the RUN/STOP/MODE dip switch and set to RUN.

5.2 CF CARD

The CF CARD LED must be solid green. If not, reinsert the CF card. The CF card must remain installed during operation of the scanner.

5.3 ERR.

If the ERR. LED is flashing, please check the multiple CPU settings.

5.4 7-Segment Display

The 7-segment display can be used to scroll the IP address and product name (refer to section 6.2). In the event of abnormal operation, the display may flash a two-digit event code (refer to Table 1). A blank display does not guarantee that the firmware is operating normally, as the event may have been cleared but not actually resolved.

5.5 Ethernet Jack

The Ethernet jack for CH1 contains two LEDs indicating link and network activity.

6. DIP Switches

6.1 RUN/STOP/MODE

The RUN/STOP/MODE dip switch must be set to RUN.

6.2 RES/SEL

Hold the RES/SEL dip switch in SEL for half a second to clear the event code and scroll the IP address and product name on the 7-segment LED display.

7. Event Codes

Table 1: Event Code List

Event Code (hexadecimal)	Event Description	Corrective Action
0	No error	None.
18	Failed to open communication with the PLC	Ensure that the dip switch is set to RUN mode.
19	Failed to get communication info from the PLC	Ensure that the dip switch is set to RUN mode.
1A	Invalid CCPU mode	Ensure that the dip switch is set to RUN mode.
1B	EtherNet/IP explicit messaging interrupt failed	Ensure that the dip switch is set to RUN mode.
1C	I/O data consumption failed due to invalid mode or consume address	Verify that the consume sizes and addresses do not exceed the maximum shared memory address of the scanner. Ensure that the dip switch is set to RUN mode.
1D	I/O data production failed due to invalid mode or produce address	Verify that the produce sizes and addresses do not exceed the maximum shared memory address of the sequencing PLC. Ensure that the dip switch is set to RUN mode.
1E	Failed to open communications to the PLC device types	Ensure that the dip switch is set to RUN mode.
1F	Failed to write to the PLC device type	Verify that the explicit messaging payload address is valid. Check the source/destination device type and point and the number of bytes in the source/destination payload.
20	Failed to read from the PLC device type	Verify that the explicit messaging payload address is valid. Check the source/destination device type and point and the number of bytes in the source/destination payload.
21	CompactFlash card is not mounted	Mount the CompactFlash card into the Q12DCCPU. Reformat the CompactFlash card if necessary.
22	Internal error.	Contact ICC for assistance.
23	Internal error.	Contact ICC for assistance.
24	Internal error.	Contact ICC for assistance.
25	Internal error.	Contact ICC for assistance.
26	Internal error.	Contact ICC for assistance.
27	Internal error.	Contact ICC for assistance.
28	Internal error.	Contact ICC for assistance.

29	Internal error.	Contact ICC for assistance.
2A	Internal error.	Contact ICC for assistance.
2B	Internal error.	Contact ICC for assistance.
2C	Internal error.	Contact ICC for assistance.
2D	Failed to initialize network interface	Ensure that the dip switch is set to RUN mode.
2E	Failed to set IP address	Ensure that the dip switch is set to RUN mode.
2F	Failed to set Subnet mask	Ensure that the dip switch is set to RUN mode.
30	Failed to set Gateway address	Ensure that the dip switch is set to RUN mode.
31	Invalid file size	Check that the EIPC_10770A.XML file does not exceed 2258432 bytes. Check that the LOG_10770A.XML file does not exceed 406035 bytes. If both files are valid, the file system may be corrupt. Reformat the CompactFlash card.
32	Failed to read file	Ensure that the CompactFlash card is mounted. Reformat the CompactFlash card if necessary.
33	Failed to write file	Ensure that the CompactFlash card is mounted. Reformat the CompactFlash card if necessary.
34	Failed to delete file	Ensure that the CompactFlash card is mounted. Reformat the CompactFlash card if necessary.
35	Internal error.	Contact ICC for assistance.
36	Internal error.	Contact ICC for assistance.
37	Failed to load EtherNet/IP configuration	The EIPC_10770A.XML may be invalid or corrupt. Verify that the Multiple CPU Settings match the settings of the sequencing PLC in slot 1. It may be necessary to default the Multiple CPU Settings. Refer to 10.10. Reformat the CompactFlash card if necessary.
38	Internal error.	Contact ICC for assistance.
39	EtherNet/IP shutting down IP	None.
3A	Internal error.	Contact ICC for assistance.
3B	CIP connection established	None.
3C	CIP connection timed out	The scanner or server device may be overloaded with incoming packets. Ensure that the system network traffic does not exceed the network bandwidth limitations. Employ IGMP enabled switches to route multicast packets more efficiently. The RPI for a connection may be increased to reduce network traffic.

3D	Forward Open request failed	The forward open request failed to establish a CIP connection with the server device. View the event history of the scanner and lookup the general status code and extended status code. Refer to 9.3.
3E	Packet interval mismatch	The requested packet interval does not match the actual packet interval. The connection will adjust to using the actual packet interval. Consult the server device vendor for appropriate packet intervals.
40	Internal error.	Contact ICC for assistance.
41	Socket connection established	None.
42	Socket connection could not be established	The scanner failed to establish a TCP connection with the server device. Ensure that both the scanner and server device are configured with compatible network settings. Check that the server device is online and configured to support EtherNet/IP. If the EIP4CCPU was previously communicating before rebooting, the EtherNet/IP sockets and connections may not have been properly closed.
43	Socket connection closed or failed	Ensure that the server device is online and communicating.
44	Internal error.	Contact ICC for assistance.
45	Internal error.	Contact ICC for assistance.
46	Internal error.	Contact ICC for assistance.
47	Failed to create log on CompactFlash card	Ensure that the CompactFlash card is mounted. Reformat the CompactFlash card if necessary.
48	Internal error.	Contact ICC for assistance.
49	Internal error.	Contact ICC for assistance.
4A	Internal error.	Contact ICC for assistance.
4B	Time trial has expired. Please obtain a valid license file.	The time trial for using the EIP4CCPU firmware has expired. A valid license file must be obtained from ICC to activate the firmware for unlimited usage.
4C	CCPU slot conflict	Check that the CPU Number is set correctly for the desired Sequencing PLC.
4D	Failed to synchronize CPU settings with sequencing PLC	Verify that the Multiple CPU Settings match the settings of the sequencing PLC in slot 1.
4E	Invalid license file. Please obtain a valid license file.	Contact ICC for a valid license file.
4F	Incompatible CCPU firmware.	The Q12DCCPU firmware is incompatible with the EIP4CCPU firmware. Refer to page 2. Contact Mitsubishi for further assistance.
50	Multiple CPU settings have changed. The CCPU must be reset.	The Q12DCCPU has been reprogrammed with the Multiple CPU settings of the sequencing PLC in slot 1 and must be reset for the settings to take effect.

51	CPU settings contain an invalid base address or size.	Verify that the Multiple CPU Settings match the settings of the sequencing PLC in slot 1.

8. Software Specifications

8.1 Overall

Item	Note
Number of client TCP connections	32 (See note 1)
Number of server TCP connections	8
PLCs supported on backplane	1 to 3
Changing configuration during operation	Yes (See note 2)

8.2 Client Class 1 Implicit (I/O) Messaging

Item	Note
CIP connections	60 (See note 3)
Total combined input and output data size	Up to 14KB (high speed shared memory limitation)
Max data size	511 bytes
Connection type (Target to Originator)	Multicast (See note 4)
Transport trigger	Cyclic
Data type	SINT, USINT
RPI	1 to 8388ms
Minimum timeout time	128ms (timeout multiplier is adjusted according to the RPI)

8.3 Client Explicit Messaging

Item	Note
Class 3 CIP connections (Connected)	16 (See note 3 and note 5)
UCMM (Unconnected)	16 outstanding requests (See note 5)
Max data size	120 words
Connection type (Target to Originator)	Point-to-Point
Cache Type	Un-cached
Transport trigger	Application

Data type	INT, UINT
RPI	7500ms
Timeout multiplier	4x
Tag access methods	Data table read/write, Typed read/write, CIP generic
PLC Implementation	Based on populating internal PLC registers with a predefined messaging structure

8.4 Server Class 3 Explicit Messaging

Item	Note
CIP connections	16
Max data size	250 words
Connection type (Target to Originator)	T->O Point-to-Point
Transport trigger	Application
Data type	INT, UINT
RPI	1 to 8388ms
Timeout multiplier	4x to 512x
Tag access methods	Data table read/write, Typed read/write, CIP generic
PLC Implementation	Predefined device mappings

Note 1: The client TCP and CIP connections share a common resource pool. The number of TCP and CIP connections is dependent upon one another and must satisfy the following formulas:

1. (Number of TCP connections * 4) + (Number of CIP connections) <= 160
2. (Number of CIP connections) <= 60

Note 2: If configuration is changed via EtherNet/IP by editing the Connection Configuration objects (RSNetWorx for EtherNet/IP method), the configuration will take effect immediately without rebooting the device. If the configuration is transferred as an XML file via FTP to the device (Windows® Configuration Utility method), a reboot is required for the configuration to take effect.

Note 3: The number of simultaneous class 1 connections lists the total number of simultaneous I/O connections that can be made to remote devices, regardless of whether or not those devices are being served by a single adapter at one IP address (modular devices such as Flex I/O and Point I/O will consume one class 1 connection for each module attached to the chassis/adapter). Because the scanner supports up to 128 TCP connections (sockets), up to 128 simultaneous physical remote devices (adapters) can be attached to. At the same time, the scanner contains a pool of 256 class 1 CIP connections and 16 class 3 CIP connections. So, for example, if the scanner is configured to target one Flex I/O adapter with three modules on the chassis, then it will use the following internal resources: one TCP connection and three class 1 CIP connections. Additionally, if the user wants a connected explicit messaging request to target (for example) module #2 on the chassis, then this will consume one of the 16 class 3 CIP connections (so in total, one TCP connection, three class 1 CIP connections, and one class 3 CIP connection will be in use). During configuration, the user can keep adding connections until they run out of either TCP connections or

class 1 connections. For example, if the network contains a large number of modular devices with multiple modules on each adapter, then it is likely that the pool of class 1 CIP connections will be exhausted first. On the other hand, if the network does not contain any modular devices, then only one class 1 CIP connection will be required for each TCP connection, and it is therefore likely that the number of TCP connections will be exhausted first.

Note 4: Up to 20 unique multicast addresses are supported per TCP connection.

Note 5: Both connected and unconnected explicit messaging requires the use of interrupts. Each interrupt can only service one outstanding explicit message at any given time. Since there are only 16 interrupts, the total number of outstanding connected and unconnected explicit messages cannot exceed 16.

9. Protocol-Specific Information

9.1 EtherNet/IP

9.1.1 Overview

The EtherNet/IP protocol is an application-level protocol implemented on top of the Ethernet TCP/IP and UDP/IP layers. It shares its object model with ControlNet and DeviceNet through the Common Industrial Protocol (CIP). This protocol allows the transfer of data and I/O over Ethernet.

EtherNet/IP incorporates both the TCP and UDP layers of Ethernet in the transmission of data. Because TCP/IP is a point-to-point topology, EtherNet/IP uses this layer only for explicit messaging; i.e. those messages in which the data field carries both protocol information and instructions for service performance. With explicit messaging, nodes must interpret each message, execute the requested task and generate responses. These types of messages can be used to transmit configuration, control and monitor data.

The UDP/IP protocol layer, which has the ability to multi-cast, is used for implicit (I/O) messaging. With I/O messaging, the data field contains only real-time I/O data; no protocol information is sent because the meaning of the data is pre-defined at the time the connection is established, which in turn minimizes the processing time of the node during run-time. I/O messages are short and have low overhead, which allows for the time-critical performance needed by controllers.

Some other notes of interest are:

- The scanner supports the EtherNet/IP protocol (release 1.0), administered by the Open DeviceNet Vendor Association (ODVA).
- This product has been self-tested by ICC, Inc. and found to comply with ODVA EtherNet/IP Conformance Test Software Version A-6.
- The scanner's product type code is 12 (Communications Adapter).
- EtherNet/IP server supports unconnected messages.
- EtherNet/IP client can simultaneously support class 1 (I/O) and class 3 (explicit) connections per TCP connection.
- EtherNet/IP client supports modular I/O (i.e. FLEX I/O, POINT I/O, etc.)
- The class 1 and class 3 connections operate independently of each other. This applies to both the server and client drivers.
- If a class 1 connection's consuming half (O→T) times out, then the producing half (T→O) will also time-out and will stop producing.
- For class 1 client connections, if the **API** (actual packet interval) returned by the server is different than the client's configured **RPI** setting, then the client will attempt to adjust to the API. This event is logged in order to notify the user that the server's data rate capabilities differ from the configured client settings. The connections will produce and consume data according to the server's **API**.
- Refer to Mitsubishi for example programs.

9.1.2 EtherNet/IP Client

The EtherNet/IP client supports both I/O and explicit messaging and can communicate with both EtherNet/IP server and CSP (PCCC) server devices.

9.1.2.1 Implicit I/O Messaging

- Configured with ESCU. Refer to section 10.
- Alternatively configured with RSNetWorx for EtherNet/IP (V10.00.00 CPR 9 SR 2). Refer to section 12.

- RSNetWorx does not have the ability to set the CPU number, shared memory base address, and shared memory sizes. Refer to section 10.10 on how to default the multiple CPU settings.
- I/O data is seamlessly transferred between the PLC and scanner via high speed shared memory. Refer to section 10.11.2 for more information.

9.1.2.2 Explicit Messaging

Up to 16 simultaneous outstanding explicit messages are supported. The following explicit messaging commands are supported:

- Get Attribute Single – Returns the contents of the specified attribute.
- Set Attribute Single – Modifies the contents of the specified attribute.
- Data Table Read – Reads data associated with the tag name.
- Data Table Write – Writes data associated with the tag name.
- Typed Read – Reads a block of data from the file number and offset.
- Typed Write – Writes a block of data to the file number and offset.

9.1.2.2.1 Explicit Messaging Structure

The ESCU will allow the user to define the sequencing PLC internal register start locations for up to 16 explicit messaging “control blocks”. This would allow up to 16 simultaneous explicit messaging transactions to occur. The general structure and attributes of these explicit messaging control blocks is provided in Table 2. A flow chart illustrating the use and processing of explicit messaging is provided in Figure 3.

Table 2: Explicit Messaging Structure

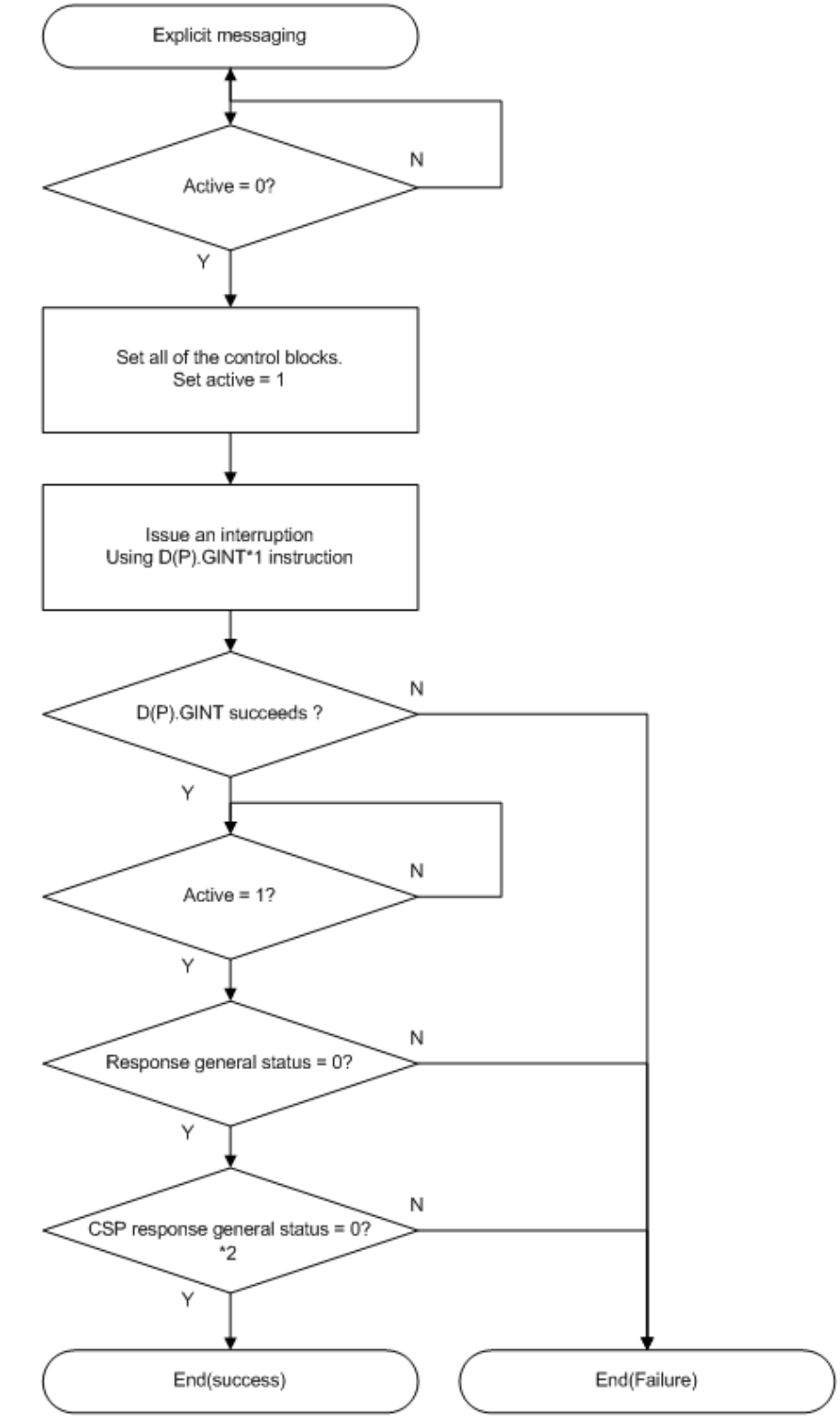
Data Type	Member Name	Get/Set Attribute Single	Data Table Read/Write	Typed Read/Write
INT16U	Version	IN	IN	IN
INT16U	Active	IN/OUT	IN/OUT	IN/OUT
INT16U	Response general status	OUT	OUT	OUT
INT16U	Response extended status	OUT	OUT	OUT
INT16U	CSP response general status	-	-	OUT
INT16U	CSP response extended status	-	-	OUT
INT16U	Service code	IN	IN	IN
INT32U	IP Address	IN	IN	IN
INT16U	Port	IN	IN	IN
INT16U	Slot	IN	IN	IN
INT8U[32]	Tag name	-	IN	-
INT16U	File number	-	-	IN
INT16U	Tag/File offset	-	IN	IN
INT16U	Class	IN	-	-
INT16U	Instance	IN	-	-
INT16U	Attribute	IN	-	-

INT16U	Source/Destination device type	IN	IN	IN
INT32U	Source/Destination device point	IN	IN	IN
INT16U	Number of bytes in source/destination payload	IN	IN	IN
INT16U	Unconnected	IN	IN	IN
INT16U[27]	Reserved (Pad structure to 64 words)	-	-	-

Explanation of explicit messaging (EM) structure members:

- **Version**: The version of the explicit messaging (EM) structure. This value is used to determine the format of the EM structure. The current version is 0.
- **Active**:
    - While zero, any of the other members of the structure (class, instance, etc.) may be freely changed by the PLC (the EM object remains dormant).
    - When non-zero ("rising edge" detection), the values contained in the other structure members will be used to generate the EM client request to the EtherNet/IP server device.
    - When the scanner has fully processed the EtherNet/IP server's response and populated the other members of the structure with the response data, it will lastly return the "active" member to 0 to inform the PLC that the EM response is ready.
    - Users only have to enter a value of 1 to "trigger" the explicit messaging request. After which the "Active" member should be polled to detect a response.
- **Response general status**: as defined by the EtherNet/IP specification. 0=service completed successfully, nonzero=error. If the service code is Typed Read or Typed Write, the CSP response general status must also be checked. Refer to 9.3
- **Response extended status**: as defined by the EtherNet/IP specification (can be ignored if response general status is 0). Refer to 9.3. If the response general status is 0x1F (Vendor specific error), refer to Table 7.
- **CSP response general status**: as defined by the CSP specification. 0=service completed successfully, nonzero=error, 0xF0=see CSP response extended status. Refer to 9.3.
- **CSP response extended status**: as defined by the CSP specification (can be ignored if CSP response general status is not 0xF0). Refer to 9.3.
- **Service Code**:
    - Get Attribute Single (0x0E) – Read a single attribute value. Must specify the Class, Instance, and Attribute fields.
    - Set Attribute Single (0x10) – Write a single attribute value. Must specify the Class, Instance, and Attribute fields.
    - Data Table Read (0x4C) - Must specify the Tag name. Tag/File offset is optional and will be ignored if left blank or 0. It is expected that only 16-bit elements will be accessed, therefore an even number of bytes must be specified.
    - Data Table Write (0x4D) - Must specify the Tag name. Tag/File offset is optional and will be ignored if left blank or 0. It is expected that only 16-bit elements will be accessed, therefore an even number of bytes must be specified.
    - Typed Read (0x68) - Must specify the File number and Tag/File offset. It is highly recommended to specify a Port value and a Slot value. A Port value of 1 and Slot value of 0 is recommended if these fields are unknown. It is expected that only 16-bit elements will be accessed, therefore an even number of bytes must be specified.
    - Typed Write (0x67) - Must specify the File number and Tag/File offset. It is highly recommended to specify a Port value and a Slot value. A Port value of 1 and Slot value of 0 is recommended if these fields are unknown. It is expected that only 16-bit elements will be accessed, therefore an even number of bytes must be specified.

- **IP Address**: IP Address of target server device. Each 8-bit byte in this 32-bit variable represents one octet in the IPV4 IP address. The IP address is not checked for validity, so the user must take precautions when entering the IP address. Unmodified by the scanner.
- **Port**: This field specifies the port to be used by the adapter to route the packet to the “target” device. The port values are specific to the adapter device on a rack-based chassis. Typically, a port value of 1 will correspond to the backplane. Valid values are 0 to 14. A value of 0 must be set if not used.
- **Slot**: This field should only be used when addressing a modular device on a rack-based chassis. The slot value will indicate the location in the rack. A Port value other than 0 must be specified if the Slot value is used. Otherwise, a Port value of 0 must be set if the Slot value is not used. Valid values are 0 to 255.
- **Tag name**: An INT8U ASCII character array (16 bytes max) used to target a tag on an EtherNet/IP server device. Use of the tag name is mutually-exclusive of use of the class, instance and attribute members mentioned below. It is recommended to use only printable ASCII characters. Unused characters should be set to 0. Unmodified by the scanner.
- **File number**: A file number. Unmodified by the scanner.
- **Tag/File offset**: An offset for the Tag name or File number. Unmodified by the scanner.
- **Class**: The targeted EtherNet/IP server device class. Unmodified by the scanner.
- **Instance**: The targeted EtherNet/IP server device instance. Unmodified by the scanner.
- **Attribute**: The targeted EtherNet/IP server device attribute. Unmodified by the scanner.
- **Source/Destination device type**: For set/write requests, this field specifies the source device type in the PLC. For get/read requests, this field is the destination device type in the PLC. Used in conjunction with the Source/Destination device point. Device types are limited to D (0x000D), W (0x0018), and ZR (0x00DC).
- **Source/Destination device point**: For set/write requests, this field is the source device point in the PLC. For get/read requests, this field is the Destination device point in the PLC. Used in conjunction with the Source/Destination device type. This mechanism allows the user to reference explicit messaging payloads of arbitrary size.
- **Number of bytes in payload**:
    - During “set/write” service codes, initially set by the PLC to indicate the number of bytes in the payload (as referenced by the “Source/Destination” device type & point attributes) to be written to the EtherNet/IP server device.
    - During “get/read” service codes, the scanner will set this value to indicate the number of bytes in the payload (as referenced by the “Source/Destination” device type & point attributes), as returned by the EtherNet/IP server device.
    - Valid values are 1 to 240 bytes. The number of bytes may have further restrictions imposed depending on the Service Code.
- **Unconnected**: If set to 0, the request will be sent using connected messaging. If the value is non-zero, the request will be sent using unconnected messaging. Unconnected messages require less overhead. Currently, unconnected messages cannot be routed using the Port and Slot values. To route the request, please use connected messaging.
- **Reserved**: These words are reserved for future expansion of the structure and will be ignored. The reserved words must pad the structure to a total size of 64 words.



*1 Refer C Controller Module User's Manual(Hardware Design, Function Explanation).
*2 When the command is Typed Read/Write.

Figure 3: Explicit Message Flow Chart

9.1.2.2.2 Explicit Message Instruction Function Block

For the user's convenience, the explicit message has been implemented as a function block (FB) using Structure Text programming language in GX Works2. Please contact Mitsubishi for more information regarding the Explicit Message Instruction Function Block.

9.1.3 EtherNet/IP Server

The EtherNet/IP server supports only explicit messaging. Class 3 (explicit messaging) can serve as a direct method of accessing the sequencing PLC's devices by reference to "tag names". Tags are read via the EtherNet/IP "Data Table Read" service, and tags are written via the EtherNet/IP "Data Table Write" service.

9.1.3.1 Device Tag Reference:

The supported device types can be accessed with their own unique tag names with one PLC instruction. Note that an underscore ("_") is used to separate fields in the tag names. Tag names are generated according to the following equation:

[device type prefix]_[address]

The address is a 32-bit unsigned value. Note that the address may take the form of a decimal or hexadecimal value depending on the device type. The supported device types and address formats are listed in the following table:

Device Type	Format (d = decimal, h = hexadecimal)
Data Register	D_<d>
Special Register	SD_<d>
File Register	ZR_<d>
Link Register	W_<h>

Examples
Access data starting at Data Register 100……D_100
Access data starting at Link Register 1000……W_3E8

9.2 Allen Bradley CSP

9.2.1 Overview

Ethernet-enabled Allen-Bradley legacy PLCs (such as the PLC5E and SLC-5/05 series) use a protocol called CSP (Client Server Protocol) to communicate over the Ethernet network. The flavor of CSP used by these PLCs is also known as "PCCC" (Programmable Controller Communication Commands) and "AB Ethernet". CSP by definitions is executed with explicit messaging. Refer to the DF1 Protocol and Command Set manual for more information on PCCC and CSP.

9.2.2 CSP Client

The CSP client follows the same behavior as the EtherNet/IP client explicit messaging. Refer to section 9.1.2.2.

9.2.3 CSP Server

The CSP server can only be accessed by explicit messaging. Similarly to the EtherNet/IP server, it can also serve as a direct method of accessing the sequencing PLC's devices by reference to a file number and offset.

9.2.3.1 File/Tag Reference

Data is read from and written to the server via CSP by reference to an integer "file/section number" and an "offset/element" within that file. Reading is performed via the CSP "PLC5 Read" (DF1 protocol Typed Read) service, and writing is performed via the CSP "PLC5 Write" (DF1 protocol Typed Write) service. The Typed Read service supports a maximum request size of 240 bytes. The Typed Write service supports a maximum of 238 bytes.

The “file/section number” is a 16-bit unsigned value and corresponds with the device type code. The “offset/element” is a 16-bit unsigned value and corresponds to the starting point address of the device. The supported device type codes are listed in Table 3.

Table 3: Supported Device Type Codes

Device Type	File/Section Number
Data Register	N13
Special Register	N14
File Register	N220
Link Register	N24

Table 4 provides some examples of various combinations of file/section numbers and offsets/elements which can be used to access devices.

Table 4: CSP Target Device Examples

Device	File/Section Number	Offset/Element
Data Register 100	N13	100
Special Register 20	N14	20
File Register 13000	N220	13000
Link Register 20000	N24	20000

9.3 CIP Status Codes

The following tables are provided for convenience and are not meant to be an all inclusive list of status codes. Codes not explicitly listed in each table are either obsolete or reserved. The provided descriptions of the codes are generic and may vary depending on the server device vendor. It is highly recommended that the user refer to the CIP specification which can be obtained from ODVA and consult the server device vendor for the appropriate status code documentation. Table 5 lists the CIP general status codes.

Table 5: General Status Codes

General Status Code (hexadecimal)	Status Name	Description of Status
00	Success	Service was successfully performed by the object specified.
01	Connection failure	A connection related service failed along the connection path. See Table 6.
02	Resource unavailable	Resources needed for the object to perform the requested service were unavailable
03	Invalid parameter value	See Status Code 0x20, which is the preferred value to use for this condition.
04	Path segment error	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.

05	Path destination unknown	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
06	Partial transfer	Only part of the expected data was transferred.
07	Connection lost	The messaging connection was lost.
08	Service not supported	The requested service was not implemented or was not defined for this Object Class/Instance.
09	Invalid attribute value	Invalid attribute data detected or error in data segment.
0A	Attribute list error	An attribute in the Get_Attribute_List or Set_Attribute_List response has a non-zero status.
0B	Already in requested mode/state	The object is already in the mode/state being requested by the service
0C	Object state conflict	The object cannot perform the requested service in its current mode/state
0D	Object already exists	The requested instance of object to be created already exists.
0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
0F	Privilege violation	A permission/privilege check failed
10	Device state conflict	The device's current mode/state prohibits the execution of the requested service.
11	Reply data too large	The data to be transmitted in the response buffer is larger than the allocated response buffer
12	Fragmentation of a primitive value	The service specified an operation that is going to fragment a primitive data value, i.e. half a REAL data type.
13	Not enough data	The service did not supply enough data to perform the specified operation.
14	Attribute not supported	The attribute specified in the request is not supported
15	Too much data	The service supplied more data than was expected
16	Object does not exist	The object specified does not exist in the device.
17	Service fragmentation sequence not in progress	The fragmentation sequence for this service is not currently active for this data.
18	No stored attribute data	The attribute data of this object was not saved prior to the requested service.
19	Store operation failure	The attribute data of this object was not saved due to a failure during the attempt.
1A	Routing failure, request packet too large	The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service.

1B	Routing failure, response packet too large	The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to abort the service.
1C	Missing attribute list entry data	The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behavior.
1D	Invalid attribute value list	The service is returning the list of attributes supplied with status information for those attributes that were invalid.
1E	Embedded service error	An embedded service resulted in an error.
1F	Vendor specific error	A vendor specific error has been encountered. The Additional Code Field of the Error Response defines the particular error encountered. Use of this General Error Code should only be performed when none of the Error Codes presented in this table or within an Object Class definition accurately reflect the error.
20	Invalid parameter	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
21	Write-once value or medium already written	An attempt was made to write to a write-once medium (e.g. WORM drive, PROM) that has already been written, or to modify a value that cannot be changed once established.
22	Invalid Reply Received	An invalid reply is received (e.g. reply service code does not match the request service code, or reply message is shorter than the minimum expected reply size). This status code can serve for other causes of invalid replies.
23	Buffer Overflow	The message received is larger than the receiving buffer can handle. The entire message was discarded.
24	Message Format Error	The format of the received message is not supported by the server.
25	Key Failure in path	The Key Segment that was included as the first segment in the path does not match the destination module. The object specific status shall indicate which part of the key check failed.
26	Path Size Invalid	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.
27	Unexpected attribute in list	An attempt was made to set an attribute that is not able to be set at this time.
28	Invalid Member ID	The Member ID specified in the request does not exist in the specified Class/Instance/Attribute
29	Member not settable	A request to modify a non-modifiable member was received

2A	Group 2 only server general failure	This error code may only be reported by DeviceNet Group 2 Only servers with 4K or less code space and only in place of Service not supported, Attribute not supported and Attribute not settable.
2B	Unknown Modbus Error	A CIP to Modbus translator received an unknown Modbus Exception Code.
2C	Attribute not gettable	A request to read a non-readable attribute was received
2D – CF		Reserved by CIP for future extensions
D0 - FF	Reserved for Object Class and service errors	This range of error codes is to be used to indicate Object Class specific errors. Use of this range should only be performed when none of the Error Codes presented in this table accurately reflect the error that was encountered. Please consult the device vendor's user manual.

Table 6 provides the extended status codes for general status code 01.

Table 6: Connection Manager Extended Status Codes

Extended Status Code (hexadecimal)	Name	Description
0100	CONNECTION IN USE OR DUPLICATE FORWARD OPEN	This extended status code shall be returned when an originator is trying to make a connection to a target with which the originator may have already established a connection
0103	TRANSPORT CLASS AND TRIGGER COMBINATION NOT SUPPORTED	A transport class and trigger combination has been specified which is not supported by the target. Routers shall not fail the connection based on the transport class and trigger combination. Only targets shall return this extended status code.
0106	OWNERSHIP CONFLICT	The connection cannot be established since another connection has exclusively allocated some of the resources required for this connection. An example of this would be that only one exclusive owner connection can control an output point on an I/O Module. If a second exclusive owner connection (or redundant owner connection) is attempted, this error shall be returned. This extended status code shall only be returned by a target node.
0107	TARGET CONNECTION NOT FOUND	This extended status code shall be returned in response to the forward_close request, when the connection that is to be closed is not found at the target node. This extended status code shall only be returned by a target node. Routers shall not generate this extended status code. If the specified connection is not found at the intermediate node, the close request shall still be forwarded using the path specified in the Forward_Close request.
0108	INVALID NETWORK CONNECTION PARAMETER	This extended status code shall be returned as the result of specifying a connection type, connection priority, redundant owner or fixed / variable that is not supported by the target application. Only a target node shall return this extended status code.

0109	INVALID CONNECTION SIZE	This extended status code is returned when the target or router does not support the specified connection size. This could occur at a target because the size does not match the required size for a fixed size connection. It could occur at a router if the requested size is too large for the specified network. Verify that the correct connection type is selected and that both the input and output sizes are appropriate for the selected connection type.
0110	TARGET FOR CONNECTION NOT CONFIGURED	This extended status code shall be returned when a connection is requested to a target application that has not been configured and the connection request does not contain a data segment for configuration. Only a target node shall return this extended status code.
0111	RPI NOT SUPPORTED	This extended status code shall be returned if the device can not support the requested O->T or T->O RPI. This extended status code shall also be used if the connection time-out multiplier produces a time-out value that is not supported by the device or the production inhibit time is not valid.
0112	RPI VALUE(S) NOT ACCEPTABLE	This error is to be returned when the RPI value(s) in the forward open request are outside the range required by the application in the target device or the target is producing at a different interval. The target shall include information with acceptable RPI(s).
0113	OUT OF CONNECTIONS	Connection Manager cannot support any more connections. The maximum number of connections supported by the Connection Manager has already been created.
0114	VENDOR ID OR PRODUCT CODE MISMATCH	The Product Code or Vendor Id specified in the electronic key logical segment does not match the Product Code or Vendor Id of the target device.
0115	PRODUCT TYPE MISMATCH	The Product Type specified in the electronic key logical segment does not match the Product Type of the target device.
0116	REVISION MISMATCH	The major and minor revision specified in the electronic key logical segment does not correspond to a valid revision of the target device.
0117	INVALID PRODUCED OR CONSUMED APPLICATION PATH	The produced or consumed application path specified in the connection path does not correspond to a valid produced or consumed application path within the target application. This error could also be returned if a produced or consumed application path was required, but not provided by a connection request.
0118	INVALID OR INCONSISTENT CONFIGURATION APPLICATION PATH	An application path specified for the **configuration data** does not correspond to a **configuration application** or is inconsistent with the consumed or produced application paths. For example the connection path specifies **float configuration data** while the **produced or consumed paths** specify **integer data**.

0119	NON-LISTEN ONLY CONNECTION NOT OPENED	Connection request fails since there are no non-listen only connection types currently open. The extended status code shall be returned when an attempt is made to establish a listen only connection type to a target, which has no non-listen only connection already established.
011A	TARGET OBJECT OUT OF CONNECTIONS	The maximum number of connections supported by this instance of the target object has been exceeded.
011B	RPI IS SMALLER THAN THE PRODUCTION INHIBIT TIME	The Target to Originator RPI is smaller than the Target to Originator Production Inhibit Time.
0203	CONNECTION TIMED OUT	This extended status code shall occur when a client tries to send a connected message over a connection that has been timed-out. This extended status code shall only occur locally at the producing node.
0204	UNCONNECTED REQUEST TIMED OUT	The Unconnected Request Timed Out error shall occur when the UCMM times out before a reply is received. This may occur for an Unconnected_Send, Forward_Open, or Forward_Close service. This typically means that the UCMM has tried a link specific number of times using a link specific retry timer and has not received an acknowledgement or reply. This may be the result of congestion at the destination node or may be the result of a node not being powered up or present. This extended status code shall be returned by the originating node or any intermediate node.
0205	PARAMETER ERROR IN UNCONNECTED REQUEST SERVICE	
0206	MESSAGE TOO LARGE FOR UNCONNECTED_SEND SERVICE	This shall be caused when the Unconnected_Send is too large to be sent out on a network.
0207	UNCONNECTED ACKNOWLEDGE WITHOUT REPLY	The message was sent via the unconnected message service and an acknowledge was received but a data response message was not received.
0301	NO BUFFER MEMORY AVAILABLE	The extended status code shall occur when insufficient connection buffer memory is available in the target or any router devices. Routers and target nodes shall return this error.
0302	NETWORK BANDWIDTH NOT AVAILABLE FOR DATA	This extended status code shall be returned by any device in the path that is a producer and can not allocate sufficient bandwidth for the connection on its link. This can occur at any node. This can only occur for connections that are specified as scheduled priority.
0303	NO CONSUMED CONNECTION ID FILTER AVAILABLE	Any device in the path that contains a link consumer for the connection and does not have an available consumed_connection_id filter available shall return this extended status code.

0304	NOT CONFIGURED TO SEND SCHEDULED PRIORITY DATA	If requested to make a connection that specifies scheduled priority, any device that is unable to send packets during the scheduled portion of the network update time interval shall return this extended status code. For example, on ControlNet this code shall be returned by a node whose MAC ID is greater than maximum scheduled node (SMAX).
0305	SCHEDULE SIGNATURE MISMATCH	This extended status code shall be returned when the connection scheduling information in the originator device is not consistent with the connection scheduling information on the target network.
0306	SCHEDULE SIGNATURE VALIDATION NOT POSSIBLE	This extended status code shall be returned when the connection scheduling information in the originator device can not be validated on the target network.
0311	PORT NOT AVAILABLE	A Port specified in a Port Segment is Not Available or does not exist.
0312	LINK ADDRESS NOT VALID	Link Address specified in Port Segment Not Valid This extended status code is the result of a port segment that specifies a link address that is not valid for the target network type. This extended status code shall not be used for link addresses that are valid for the target network type but do not respond.
0315	INVALID SEGMENT IN CONNECTION PATH	Invalid Segment Type or Segment Value in Connection Path This extended status code is the result of a device being unable to decode the connection path. This could be caused by an unrecognised path type, a segment type occurring unexpectedly, or a myriad of other problems in the connection path.
0316	FORWARD CLOSE SERVICE CONNECTION PATH MISMATCH	The connection path in the Forward_Close service does not match the connection path in the connection being closed.
0317	SCHEDULING NOT SPECIFIED	Either the Schedule Network Segment was not present or the Encoded Value in the Schedule Network Segment is invalid (0).
0318	LINK ADDRESS TO SELF INVALID	Under some conditions (depends on the device), a link address in the Port Segment which points to the same device (loopback to yourself) is invalid.
0319	SECONDARY RESOURCES UNAVAILABLE	In a dual chassis redundant system, a connection request that is made to the primary system shall be duplicated on the secondary system. If the secondary system is unable to duplicate the connection request, then this extended status code shall be returned.
031A	RACK CONNECTION ALREADY ESTABLISHED	A request for a module connection has been refused because part of the corresponding data is already included in a rack connection.

031B	MODULE CONNECTION ALREADY ESTABLISHED	A request for a rack connection has been refused because part of the corresponding data is already included in a module connection.
031C	MISCELLANEOUS	This extended status is returned when no other extended status code applies for a connection related error.
031D	REDUNDANT CONNECTION MISMATCH	This extended status code shall be returned when the following fields do not match when attempting to establish a redundant owner connection to the same target path
031E	NO MORE USER CONFIGURABLE LINK CONSUMER RESOURCES AVAILABLE IN THE PRODUCING MODULE	A target shall return this extended status when the configured number of consumers for a producing application are already in use.
031F	NO MORE USER CONFIGURABLE LINK CONSUMER RESOURCES AVAILABLE IN THE PRODUCING MODULE	A target shall return this extended status when there are no consumers configured for a producing application to use.
0800		Network link in path to module is offline
0810	NO TARGET APPLICATION DATA AVAILABLE	This extended status code is returned when the target application does not have valid data to produce for the requested connection. Only the target side of a connection shall return this extended status code.
0811	NO ORIGINATOR APPLICATION DATA AVAILABLE	This extended status code is returned when the originator application does not have valid data to produce for the requested connection. Only the originator side of a connection shall indicate this extended status code.
0812	NODE ADDRESS HAS CHANGED SINCE THE NETWORK WAS SCHEDULED	A router on a scheduled network (e.g.: ControlNet) has a different node address than the value configured in the connection originator.
0813	NOT CONFIGURED FOR OFF-SUBNET MULTICAST	A multicast connection has been requested between a producer and a consumer that are on different subnets, and the producer is not configured for off-subnet multicast.
0814	INVALID PRODUCE/CONSUME DATA FORMAT	Information in the data segment indicates that the format of the produced and/or consumed data is not valid.

After initiating an explicit message request from the scanner, if the response general status is 0x1F, the response general status was generated by the scanner due to incomplete processing of the request. Refer to Table 7 for a list of the ICC vendor specific response extended status descriptions:

Table 7: ICC Vendor Specific Response Extended Status Codes

Response Extended Status (hexadecimal)	Description
0001	A connection with the target could not be established or was lost.

0002	Timed out waiting for the response.
0003	The request contains an invalid number of bytes to read or write.
0004	There are insufficient resources to complete the request.
0005	The service code is invalid.
0006	The tag name is invalid.
0007	The port id is invalid.
0008	The payload device type is invalid.
0009	Failed to read data from the device address.
000A	Failed to write response data to the device address.
000B	Received a bad or malformed response.
000C	Response contains an invalid payload size.
000D	The request specifies an invalid explicit message structure version.
000E	The active member was not set high.
000F	The slot number is invalid.
0010	The scanner cannot perform the requested service in its current mode or state.
0011	The license file is invalid. Trial mode has expired.

Table 8 lists the CSP general status codes.

Table 8: CSP General Status Codes

General Status Code (hexadecimal)	Description
00	Success
10	Illegal command or format
20	Host has a problem and will not communicate
30	Remote node host is missing, disconnected, or shut down
40	Host could not complete function due to hardware fault
50	Addressing problem, wrong station number, or memory protect rungs
60	Requested function is not available
70	Processor is in Program mode
80	Compatibility mode file missing or communication zone problem
90	Remote node cannot buffer command
A0	Wait ACK
B0	Remote node is downloading so it is not accessible
C0	Wait ACK
F0	Error code in the EXT STS byte (see Table 9)

Table 9 lists the CSP extended status codes. These codes apply when the CSP general status code is 0xF0.

Table 9: CSP Extended Status Codes

Extended Status Code (hexadecimal)	Description
01	A field has an illegal value or the address was incorrectly converted
02	Incomplete address. Less levels specified in address than minimum for any address.
03	Incorrect address. More levels specified in address than system supports.
04	Illegal address format. Symbol not found.
05	Illegal address format. Symbol is of improper format.
06	Address file doesn't point to something usable or does not exist
07	Requested size is invalid or unsupported by the file
08	Cannot complete request, situation has changed since the start of the command
09	Data or file is too large or memory is unavailable
0A	Transaction size plus word address is too large. Target processor cannot put requested information in packets.
0B	Access denied, improper privilege
0C	Condition cannot be generated. Resource is not available.
0D	Condition already exists. Resource is already available.
0E	Command cannot be executed
0F	Histogram overflow
10	No access
11	Illegal data type or data type mismatch
12	Invalid parameter or invalid data
13	Address reference exists to deleted area
14	Command execution failure for unknown reason
15	Data conversion error
16	Scanner not able to communicate
17	Type mismatch
18	Module response was not valid
19	Duplicated label
1A	File is open; another node owns it
1B	Another node is the program owner
1E	Data table element protection violation
1F	Temporary internal problem
22	Remote rack fault

23	Timeout
24	Unknown error

10. EIP4CCPU Scanner Configuration Utility

The EIP4CCPU Scanner Configuration Utility (ESCU) runs on a Windows PC and is the primary tool used to configure the EIP4CCPU EtherNet/IP scanner via Ethernet. The latest utility can be downloaded from http://www.iccdesigns.com. The utility is built on the Microsoft .NET Framework for compatibility with Windows operating systems and has been tested with Windows XP SP3, Vista, and 7. The intuitive user interface allows drag and drop operations along with auto-map of input and output data to quickly configure and manage connections to devices on the network. Configuring a scanner with the utility is as simple as adding a device, configuring the connection properties, and downloading the configuration to the scanner.

Configurations can be created offline, saved as XML configuration files, and downloaded to a scanner at a later time. Configurations can also be downloaded to connected scanners as soon as they are created.

Another feature of the utility is its ability to discover and configure the network settings of any Ethernet-enabled ICC scanner on the local subnet.

The Configuration Utility is not limited to configuration alone. The utility actively monitors the status of connected scanners. It can retrieve and display event history as well as set the real-time clock of scanners.

If initiation of class 3 explicit messaging requests is desired, the configuration utility provides the ability to enable and initially setup the class 3 explicit messaging mechanism. This must be completed before attempting to execute explicit messaging requests via GX Developer or GX Works2 (refer to section 9.1.2.2).

It is highly recommended to first review and become familiar with this section before proceeding with actual configuration of an EtherNet/IP network. The remainder of this section will first detail the installation and setup of the utility. Next is a guide for registering EDS files, discovering scanners, adding devices, configuring devices and connections, downloading and uploading configurations, and managing configuration files and projects. For more information on configuring modular devices, see section 11.

10.1 Features

- Define input, output, and configuration data exchanged between devices on the network
- Automatic mapping of input and output data
- Configuration file management of XML-formatted files
- Discover and configure network settings of any scanner on the Ethernet subnet
- Monitor a scanner's current status and view event history
- Set scanner's real-time clock
- Multi-vendor interoperability with Electronic Data Sheet (EDS)
- Value fields can be specified in decimal, hexadecimal (leading "0x"), octal (leading "0") and binary (leading "0b") formats

10.2 Installation

The Configuration Utility requires that Microsoft .NET Framework V3.5 SP1 is installed on the PC. Microsoft .NET Frameworks can be downloaded from http://www.microsoft.com. Once the appropriate .NET Framework has been verified or installed, double-click the setup.exe file to start installation of the Configuration Utility. Follow the instructions to complete installation.

Review the installation message (Figure 4) and click Next.

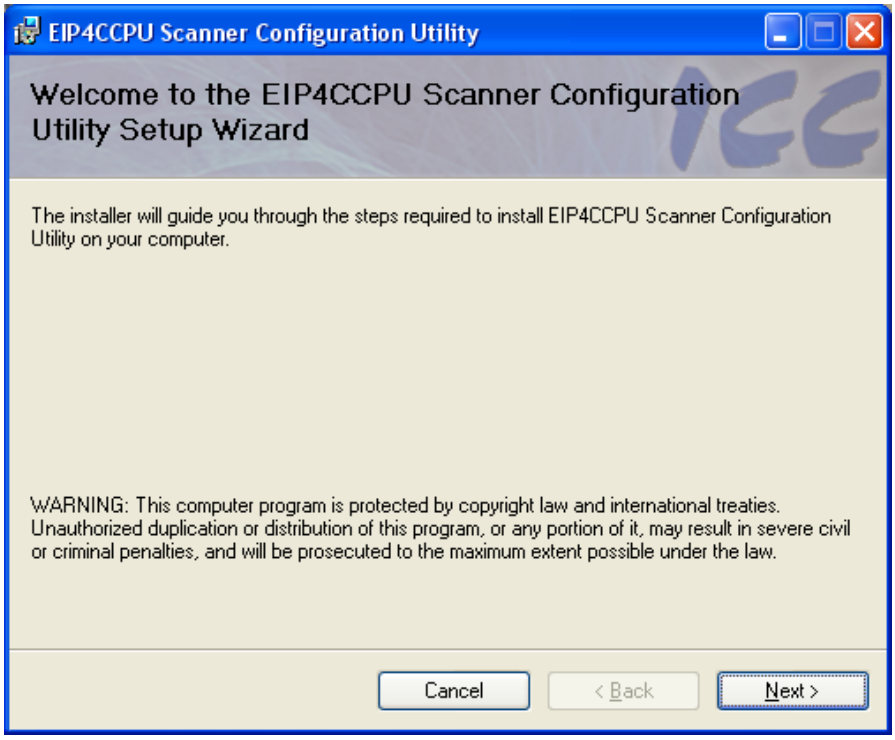


Figure 4: Installer Welcome Screen

Select the folder where you want the Configuration Utility to be installed (Figure 5).

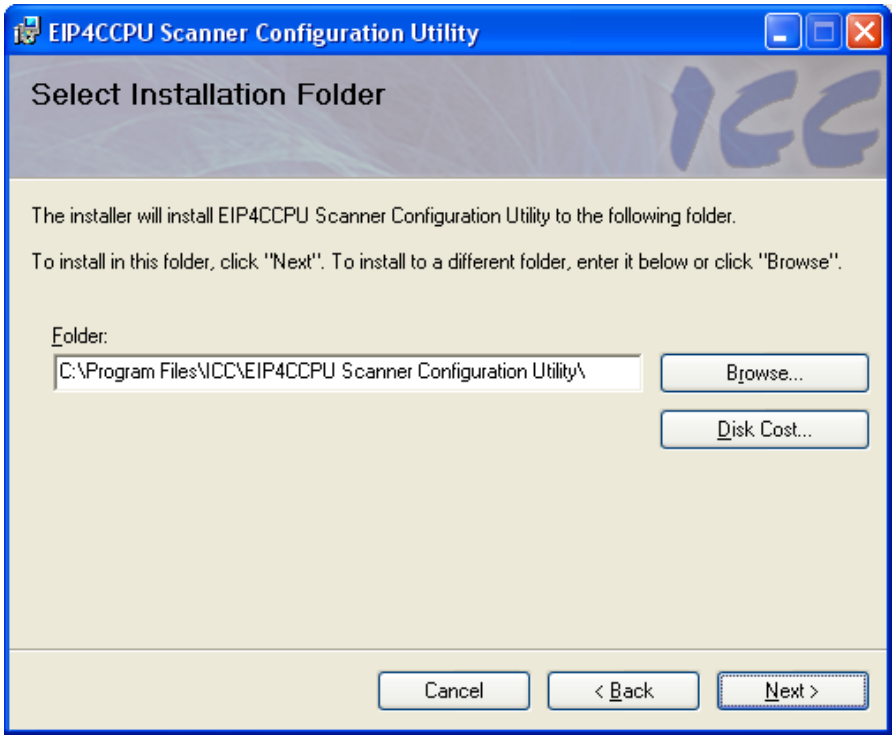


Figure 5: Installation Folder Selection Screen

Confirm and click Next to start installation (Figure 6).

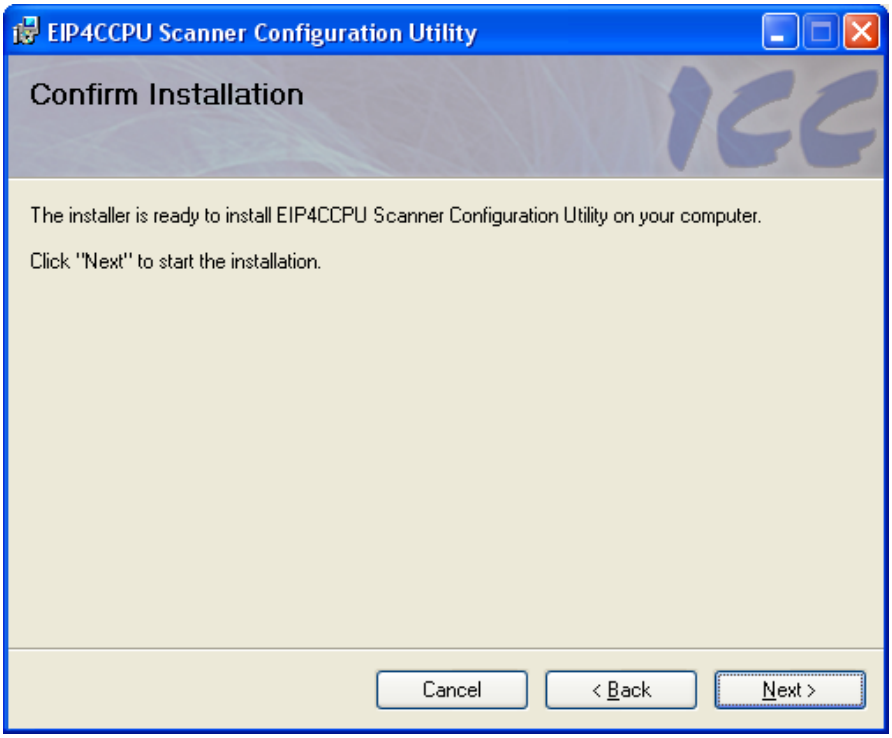


Figure 6: Installation Confirmation Screen

Wait while the Configuration Utility is being installed (Figure 7).

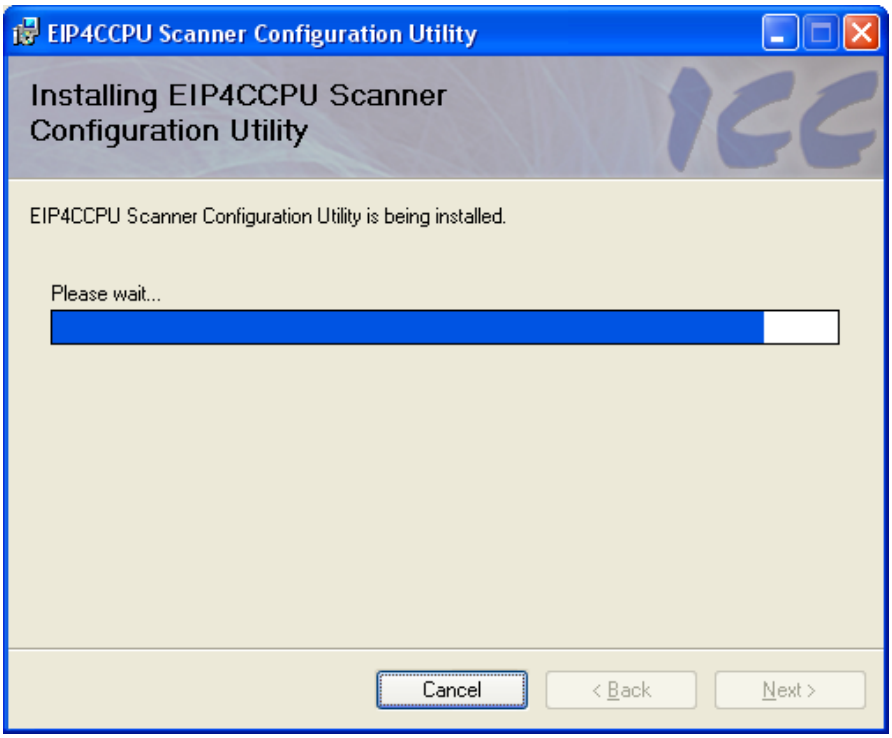


Figure 7: Installation Progress Screen

Click Close to exit the installer (Figure 8).

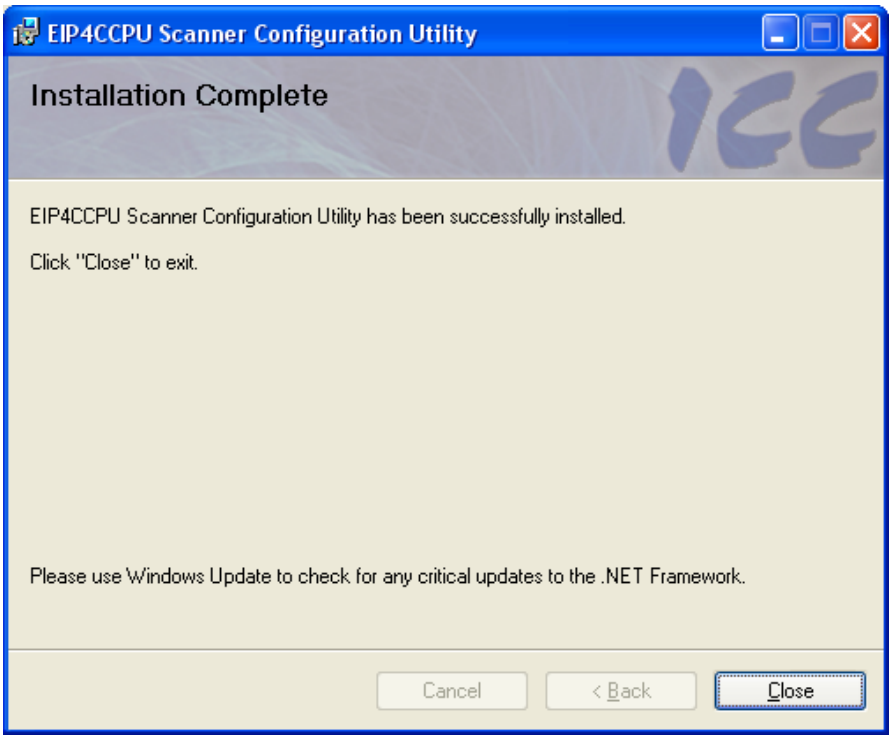


Figure 8: Installation Complete Screen

10.3 Firewall Settings

To ensure proper network communication between the configuration utility and the scanners residing on the network, please configure the PC's firewall with the following settings:

1. Add the EIP4CCPU Scanner Configuration Utility as a program exception.
2. Add UDP port 4334 as a UDP port exception.

The following procedure adds the utility and port as firewall exceptions on a Windows XP machine. Other versions of Windows will require similar steps.

Navigate to Control Panel and open Windows Firewall. Click on the Exceptions tab (Figure 9).

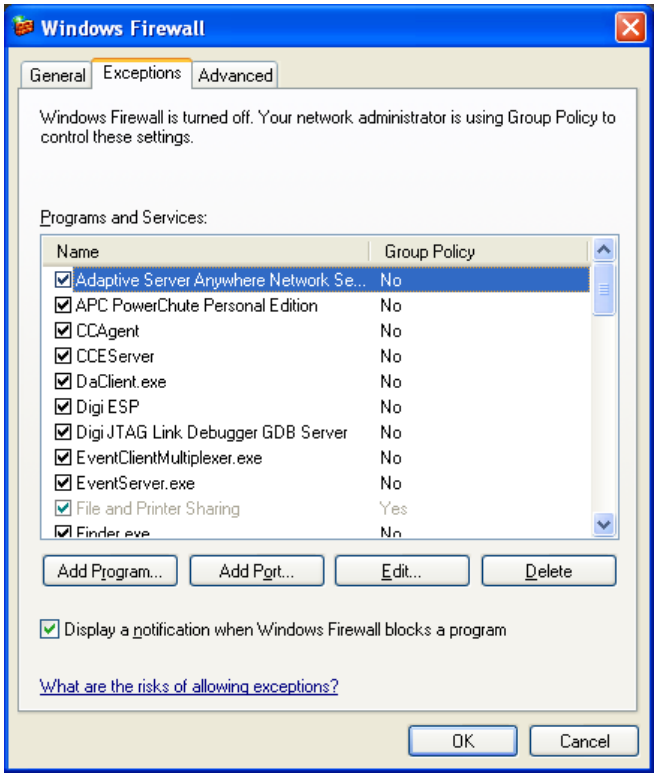


Figure 9: Windows Firewall Exceptions

Click on Add Program (Figure 10).

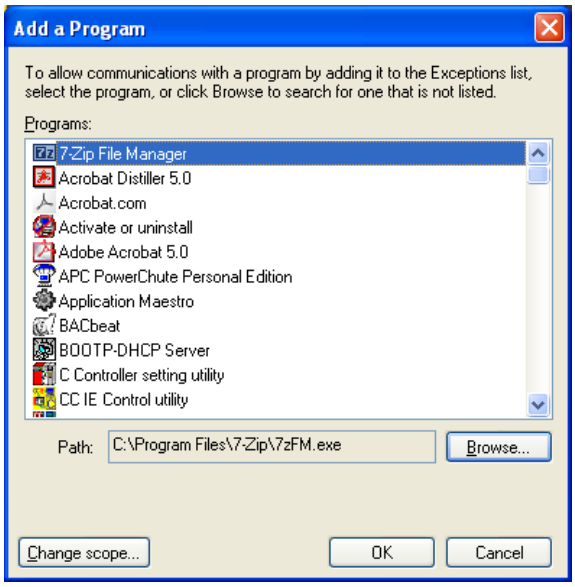


Figure 10: Adding Program to Firewall Exception

Click on Browse and locate the EIPScanner.exe executable file. The program should be highlighted in the list (Figure 11).

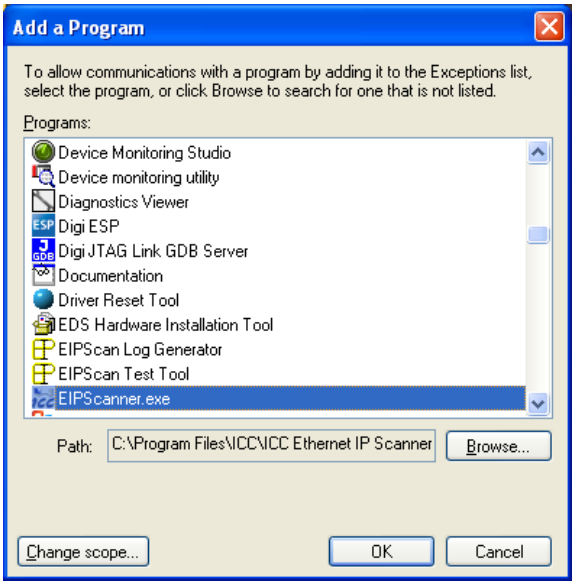


Figure 11: Utility In Program List

Click OK to add the highlighted utility to the Programs and Services exception list (Figure 12).

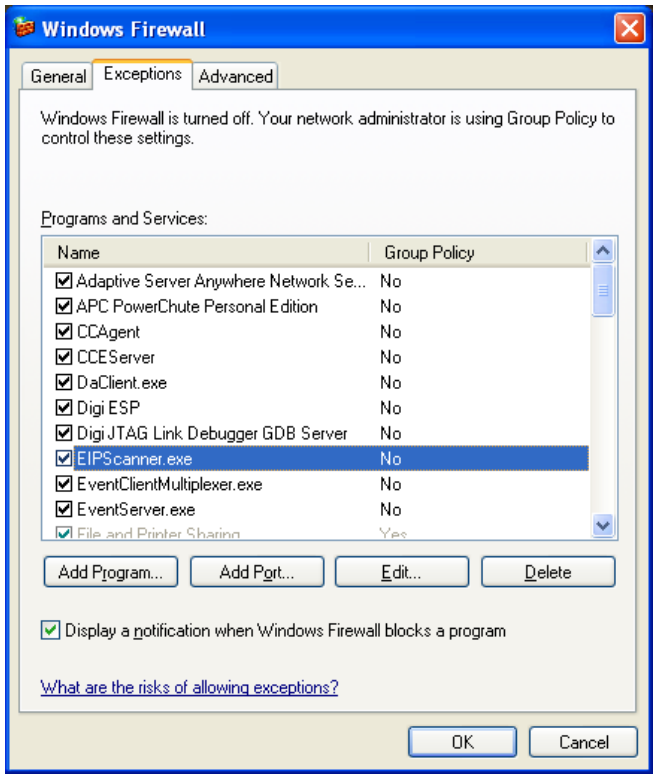


Figure 12: Added Utility to Firewall Exceptions List

Click on Add Port. In the Name field enter “IDP” and for the Port number enter “4334”. Select UDP. See Figure 13.

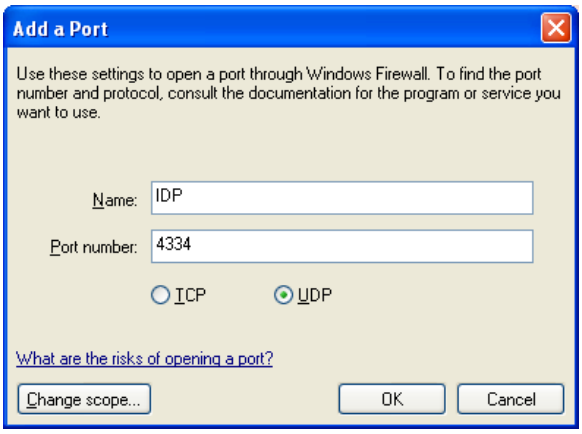


Figure 13: Adding UDP Port to Firewall Exception

Click OK. The port has been added as a firewall exception (Figure 14).

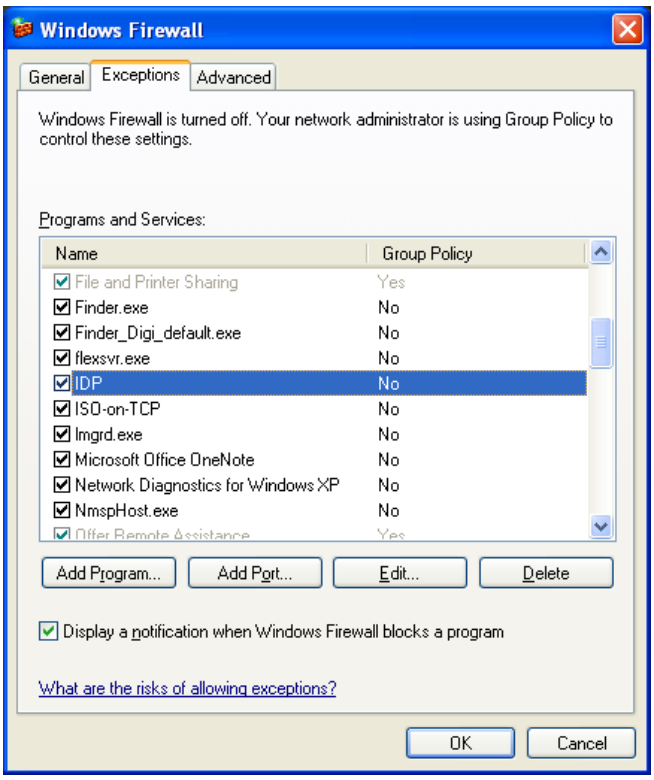


Figure 14: Added Port to Firewall Exceptions List

If the Configuration Utility is not detecting any devices after adding the exceptions to the firewall, please consult your IT to resolve the issue. Although it is not recommended, alternatively the firewall could be temporarily disabled while using the Configuration Utility.

10.4 User Interface

Refer to Figure 15.

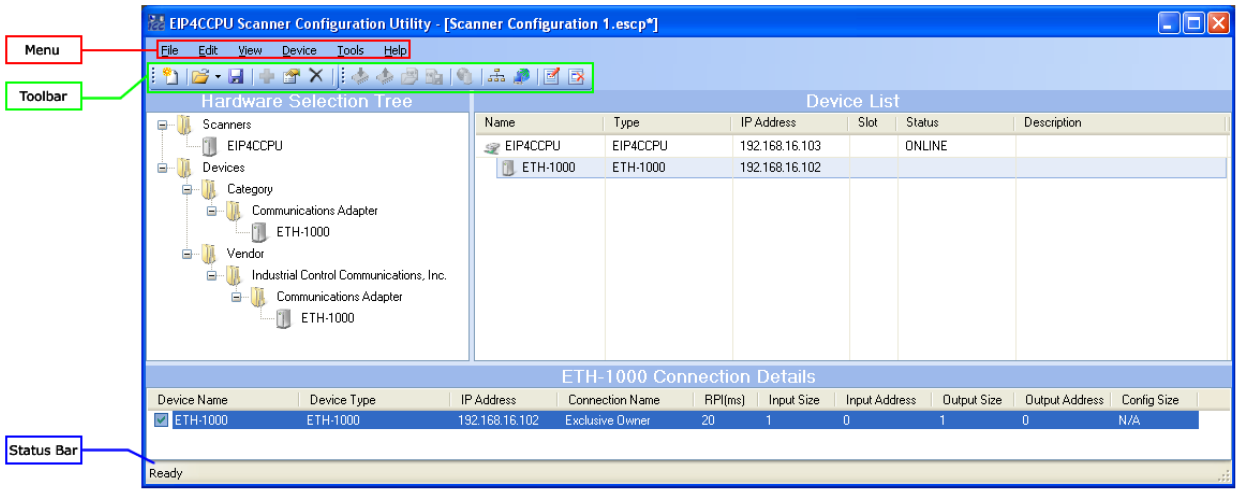


Figure 15: Utility User Interface

The ESCU user interface is broken down into 6 major elements:

1. Menu
    a. File – File management options to save, open, and create new project files.
    b. Edit –Options will vary on what item is selected. Devices can be added or deleted. Connections can be added, edited, and deleted.
    c. View – Toggle visibility of various window elements.
    d. Device – Device configuration and discovery.
    e. Tools – Register and unregister EDS files.
    f. Help – View help file or the about box.
2. Toolbar
    a. Standard – Contains shortcut icons for File and Edit menus.
    b. Actions – Contains shortcut icons for Device and Tools menus.
3. Hardware Selection Tree – Displays available devices which have been registered with the software. The tree is organized by category and vendor.
4. Device List – Displays list of devices in a tree/list structure with scanners at the roots.
5. Connection Details – Displays the details for all class 1 connections for the selected device. Selecting a scanner will display all connections to that scanner. To only display connections between one specific device and the scanner, select the desired device to filter out other connections.
6. Status Bar – Displays the status of the utility along with progress information.

The various elements of the user interface can be sized and adjusted accordingly for easy viewing.

10.5 Register EDS Files

Register the EDS files of all devices on the network. This will ensure that devices are properly recognized and configurable. Unregistered devices cannot be configured without first registering its EDS file. To overwrite an existing EDS file, register an EDS file for that device with the same Major Revision and Minor Revision. If a device has an EDS file with a different Major Revision or different Minor Revision, registering this EDS file will create an additional selection for this particular device. Use caution when registering multiple EDS files for the same device. EDS files can be obtained from the device manufacturer.

Navigate to Tools->Register EDS File (Figure 16).

Figure 16: EDS File Registration

Click Browse and select the EDS file. The ETH-1000 Ethernet gateway was selected in this example (Figure 17). Note that multiple EDS files can be selected by holding down the “Shift” button.

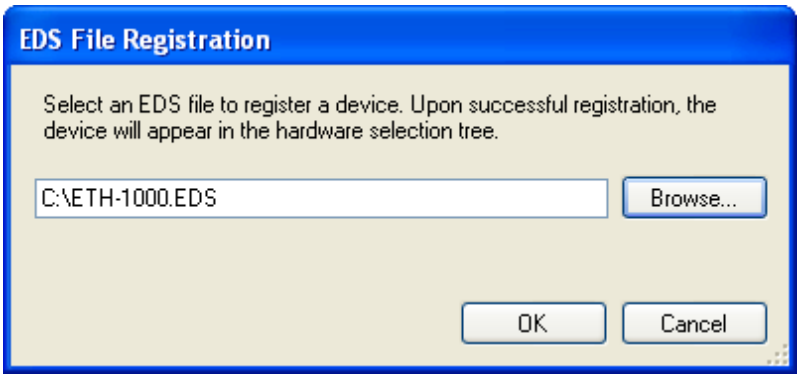


Figure 17: Select EDS File

Click OK to begin registering the EDS file. If the EDS file is successfully registered, the device will be added to the hardware tree (Figure 18).

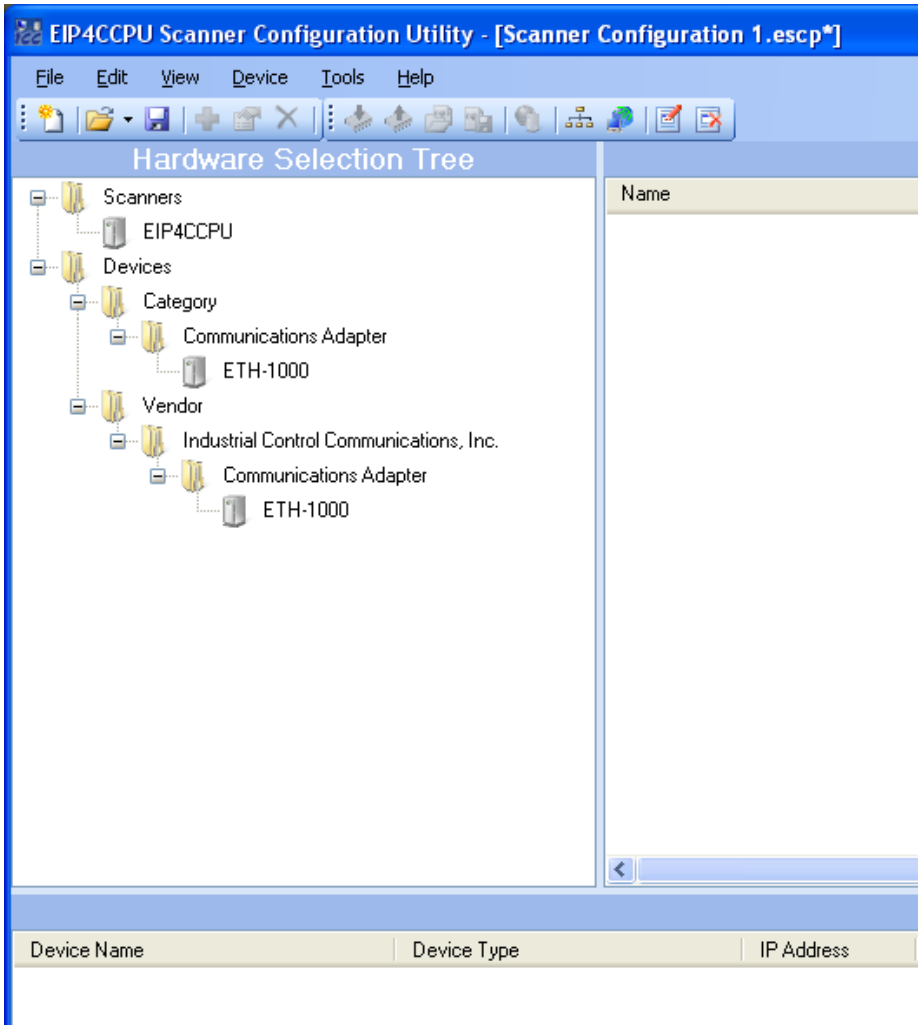


Figure 18: Device Added to Hardware Selection Tree

10.6 Unregister EDS Files

To remove devices from the Hardware Selection Tree, the EDS file of the device must be unregistered. Once a device is unregistered, it will no longer be available to be added to a Scanner in the Device List. The device's EDS file will be permanently removed from the ESCU. To restore a device, the EDS file must be registered (See 10.5).

To unregister a device's EDS file, navigate to Tools->Unregister EDS File (Figure 19). Select a device and click Unregister. Note that multiple devices can be selected by holding down the "Shift" button.

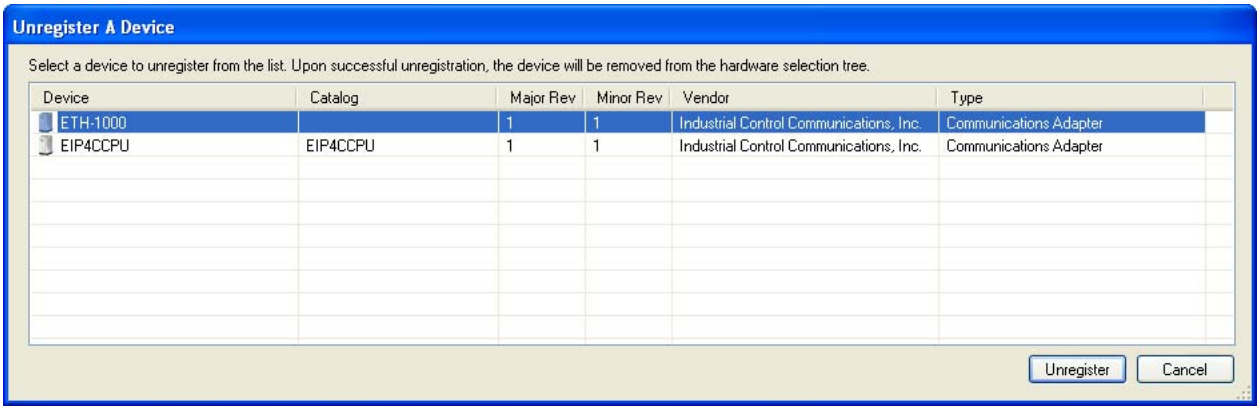


Figure 19: Unregister EDS File

10.7 Device Discovery Wizard

To discover all local scanners (regardless of their current network settings) and change their network settings, complete one of the following methods:

1. Click on the Device Discovery Wizard icon in the toolbar.
2. Navigate to Device->Device Discovery Wizard.

Figure 20 will appear. Click Next to start the scan (Figure 21).

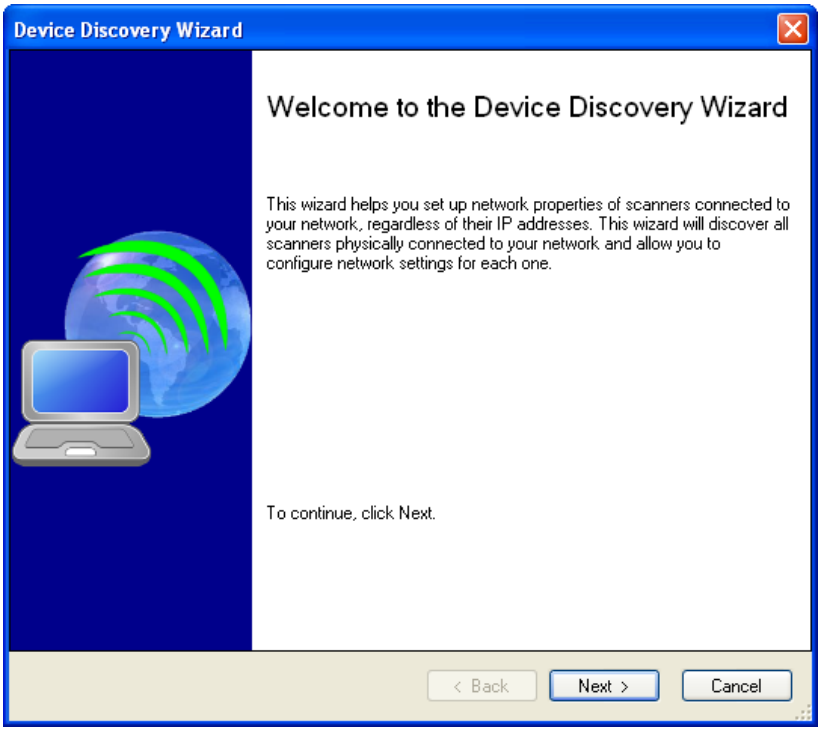


Figure 20: Device Discovery Wizard welcome screen

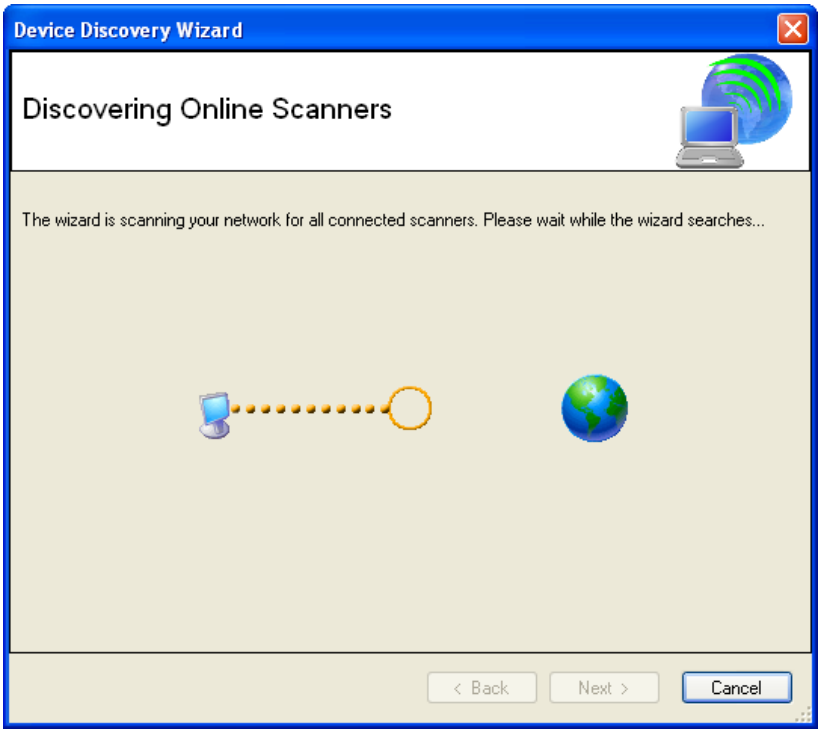


Figure 21: Device Discovery Wizard scanning network

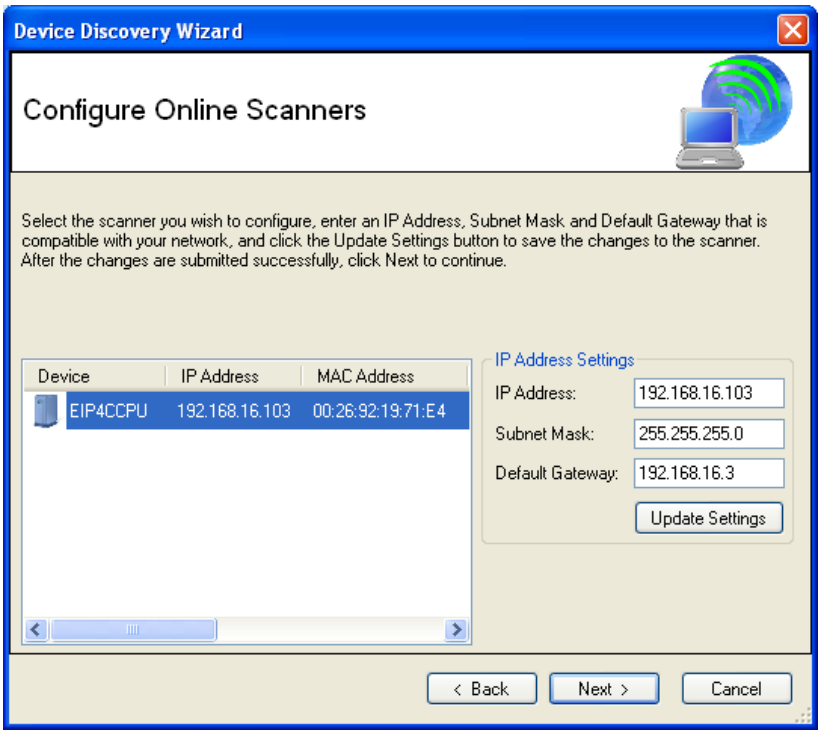


Figure 22: Discovered online scanners

Figure 22 displays the discovered scanners. Select any scanner in the list to change its IP Address Settings. Once the IP Address Settings have been modified to be compatible with the network, click Update Settings to configure the

scanner with the new IP Address Settings. The scanner must be reset for the changes to take effect (Figure 23). Once all scanners have been updated and reset, click Next and then click Finish to complete the wizard.

Figure 23: Reset device for network configuration to take effect

10.8 Discover Local Scanners

To automatically discover local scanners and upload their configurations, complete one of the following methods:

1. Click on the Discover Local Scanners icon in the toolbar.
2. Navigate to Device->Discover Local Scanners.

Applicable scanners must reside on the local subnet and have compatible network settings.

10.9 Adding a Scanner

To add a scanner, expand the Scanners menu in the Hardware Selection Tree and select EIP4CCPU (Figure 24).

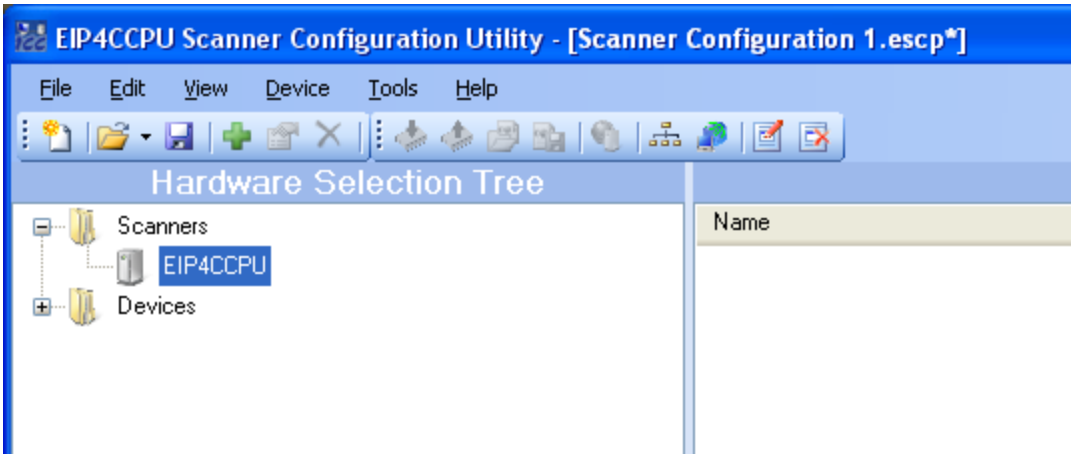


Figure 24: Select the EIP4CCPU Scanner Device

Complete one of the following methods to add the scanner to the Device List:

1. Drag and drop the EIP4CCPU onto the Device List panel
2. Double-click on EIP4CCPU
3. Right click on EIP4CCPU and Add Device
4. Click on the Add icon in the toolbar
5. Navigate to Edit->Add

Configure the IP address of the scanner (Figure 25) and click OK to add it to the Device List (Figure 26).

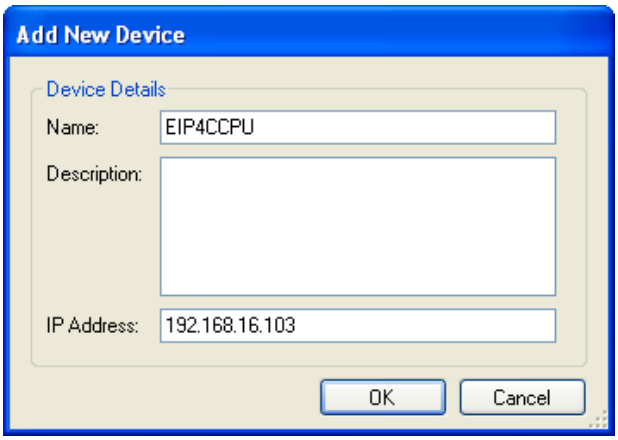


Figure 25: Add EIP4CCPU Scanner

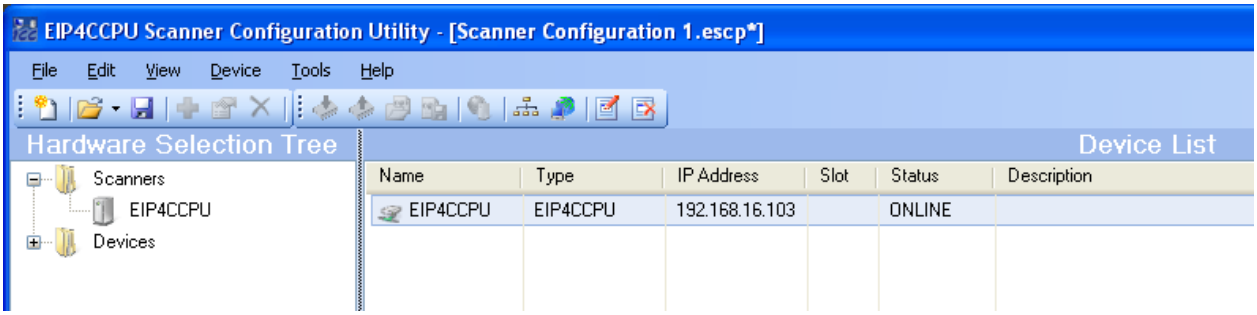


Figure 26: Added Scanner to Device List

Once a scanner has been added to the Device List, the utility will periodically check the scanner's status. If the utility can connect to the scanner (via point-to-point communications), the Status will display a status other than OFFLINE.

10.10 Default Configuration

To default the configuration, delete the configuration file "EIPC_10770.XML" from the CF card and reboot the CPU. The default configuration will only contain the following settings:

- All explicit messaging channels are disabled.
- Sequencing PLC CPU Number is set to 1.
- Sequencing PLC Shared Memory Base Address is set to 10000.
- Sequencing PLC shared Memory Size is synchronized to the multiple CPU settings of CPU number 1.
- Scanner Shared Memory Size is synchronized to the multiple CPU settings of CPU number 1.

10.11 Configuring the Scanner

It's advised to upload the scanner's existing configuration (section 10.16) before proceeding. Right click the desired scanner and select Properties or Navigate to Edit->Edit. Check all available tabs and make any necessary changes.

10.11.1 General

Refer to Figure 27.

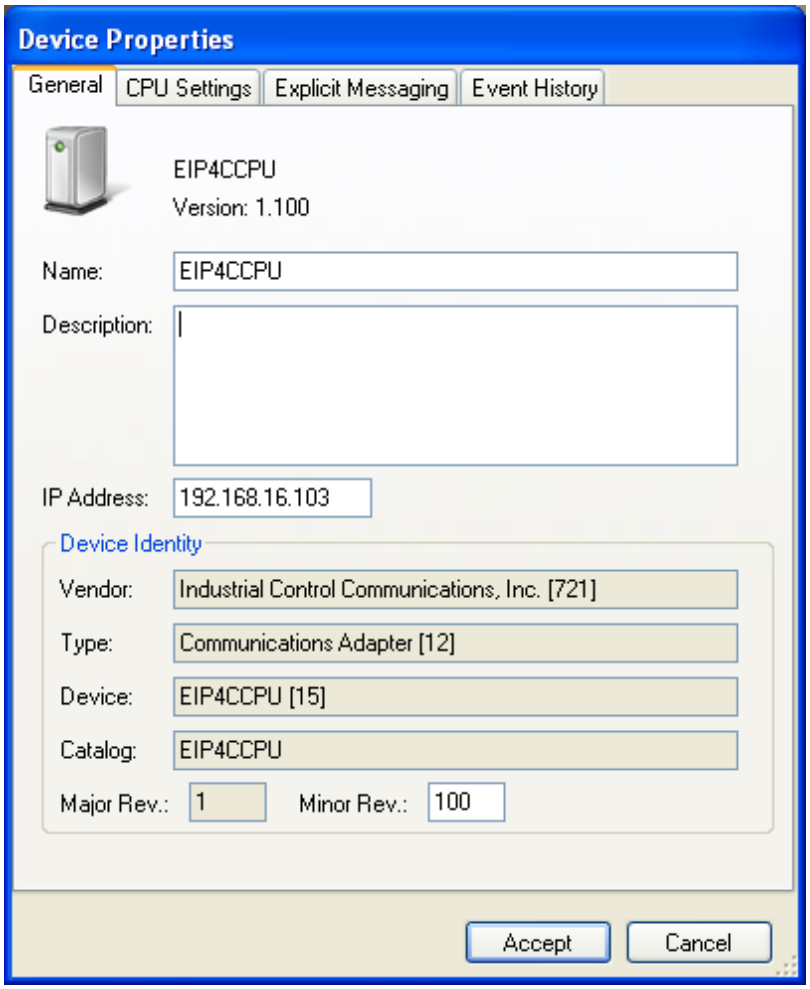


Figure 27: Scanner Device Properties General Tab

- Version – EIP4CCPU firmware version. This should not be confused with the CCPU hardware and firmware versions.
- Name – Name for the device.
- Description – Description for the device.
- IP Address – IP Address of the device.
- Device Identity – CIP identification properties.
    - Minor Rev – Minor revision should be set to match the EIP4CCPU scanner firmware minor revision.

10.11.2 CPU Settings

Refer to Figure 28.

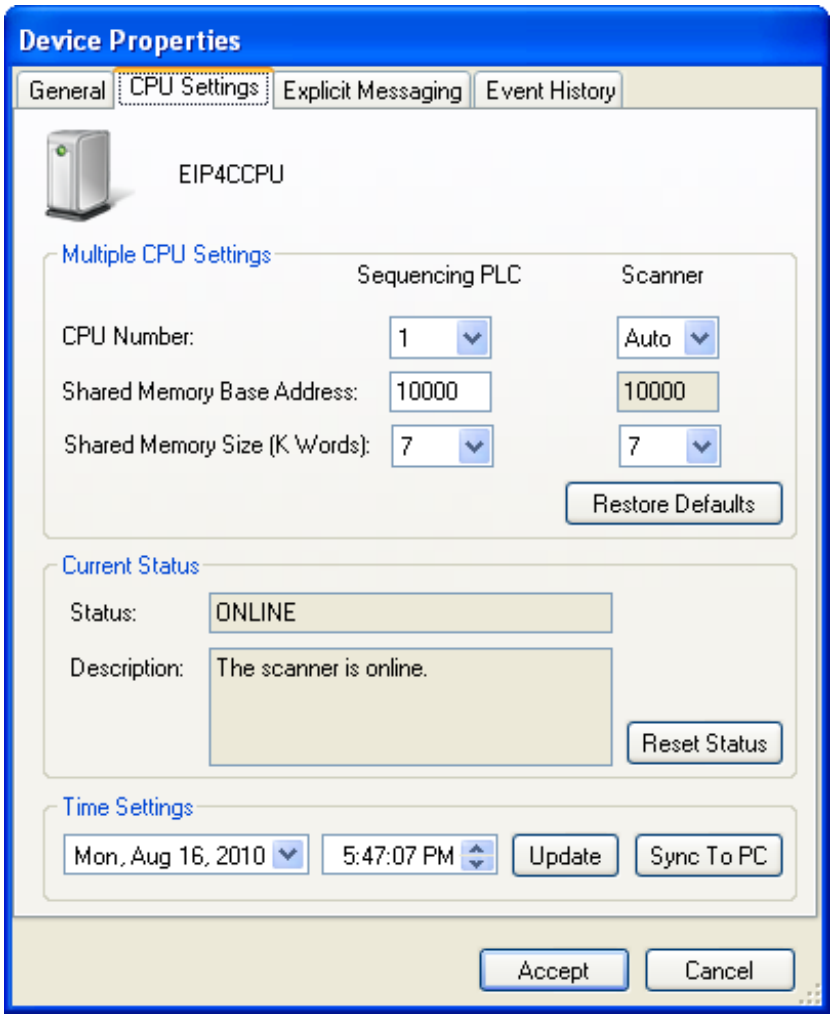


Figure 28: Scanner Device Properties CPU Settings Tab

- Multiple CPU Settings
  - CPU Number – Select the sequencing PLC to be paired with the scanner.
  - Shared Memory Base Address – Starting address of I/O data. The address must be located in high speed shared memory.
  - Shared Memory Size – Must be set according to the selected sequencing PLC multiple CPU settings. The memory size for the sequencing PLC specifies the maximum amount of output data that can be produced. The memory size for the scanner specifies the maximum amount of input data that can be consumed.

- Current Status
  - Status – Current status of the scanner.
  - Description – Description of the status code.
  - Reset Status – Click to reset/clear the status code.

- Time Settings – Real-time clock setting of the scanner.
  - Update – Click to update the scanner's real-time clock.
  - Sync To PC – Click to synchronize the scanner's real-time clock with the PC's current time.

10.11.3 Explicit Messaging

Refer to Figure 29.

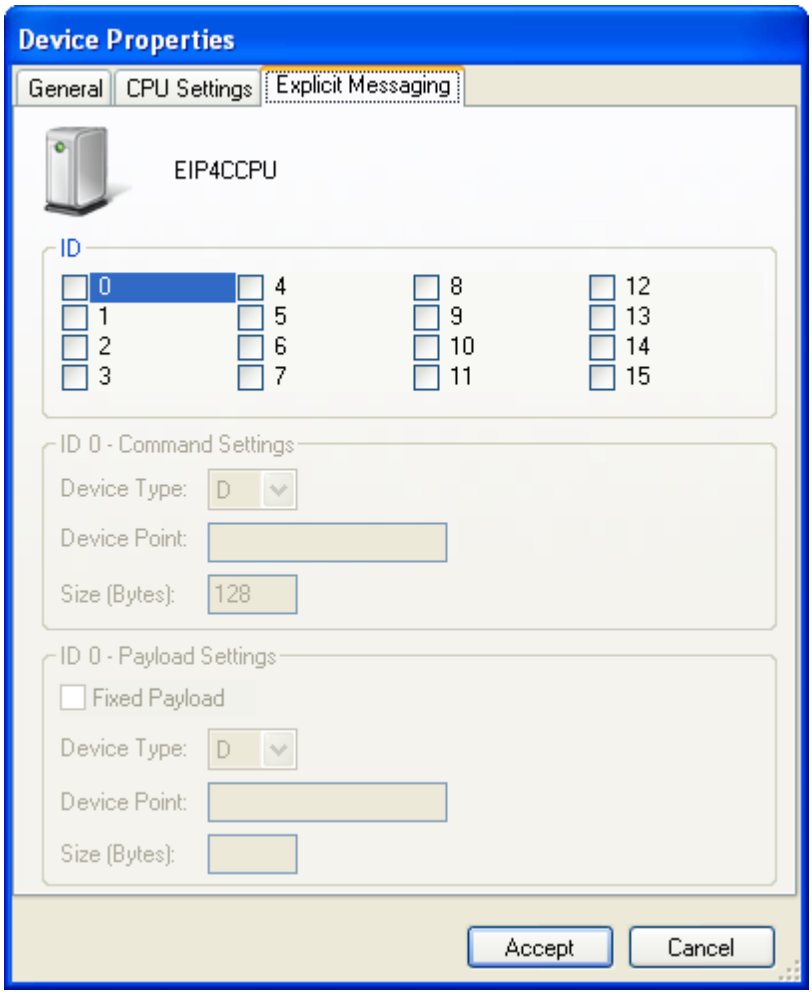


Figure 29: Scanner Device Properties Explicit Messaging Tab

- ID – The available interrupt channels. Check to enable the desired explicit messaging interrupt channel.

- Command Settings – Setup for the explicit messaging command structure.
  - Device Type – Select from D (Data Register), W (Link Register), and Z (Index Register).
  - Device Point – Used in conjunction with Device Type.
  - Size – 128 bytes (64 words) reserved for command structure and cannot be modified by the user.

- Payload Settings – Optionally hardcode the payload settings.
  - Fixed Payload – Check to setup a fixed payload location and size. If this option is set, the Source/Destination device type, Source/Destination device point**,** and Number of bytes in payload members of the explicit messaging structure will be ignored. The following Device Type, Device Point, and Size values will be used for all requests on this interrupt channel.
  - Device Type - Select from D (Data Register), W (Link Register), and Z (Index Register).
  - Device Point - Used in conjunction with Device Type.
  - Size – Number of bytes reserved for payload. If an odd number of bytes are specified, that last half word will be consumed and should not be used.

10.12 Adding a Target Device and Configuring Connection Properties

A target/server device can be added to a scanner in the Device List. Before adding a target/server device, first select the desired scanner. Next locate and select the target/server device in the Hardware Selection Tree. Adding a target/server device to a scanner can be completed with one of the following procedures:

1. Drag and drop the device to the scanner in the Device List.
2. Double-click on the device.
3. Right click on the device and select Add Device.
4. Select the device and click on the Add icon in the toolbar.
5. Select the device and navigate to Edit->Add.

The Name and Description fields can optionally be modified. Set the IP Address and click Next. Refer to Figure 30.

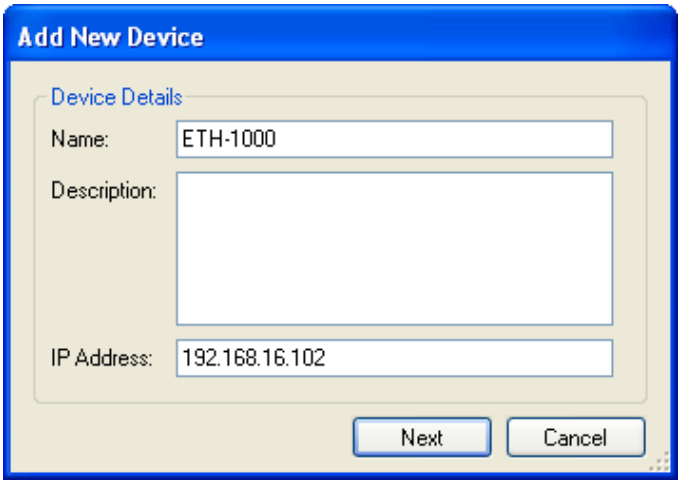


Figure 30: Adding a New Device to the Scanner

Review and configure the Connection Properties. The Connection Properties are configured with the default settings as specified by the device EDS file. Please consult the device vendor for more information regarding configuration data. Refer to Figure 31, Figure 32, and Figure 33.

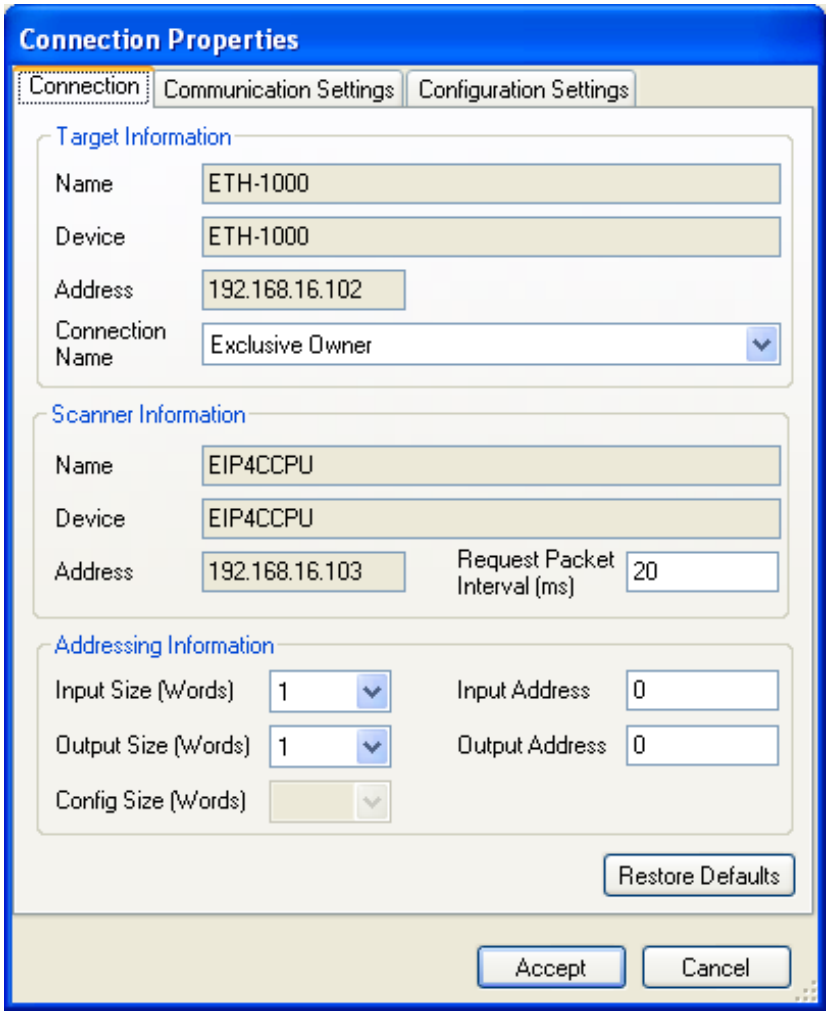


Figure 31: Connection Properties Connection Tab

- Target Information
  - Connection Name – Select the desired connection.

- Scanner Information
  - Request Packet Interval (RPI) – The rate at which I/O data is exchanged. If the target device responds with a different packet interval rate, the scanner will attempt to satisfy the target's packet interval rate.

- Addressing Information
  - Input Size – The number of words (16-bit values) that are consumed by the scanner.
  - Input Address – The offset from the scanner's Shared Memory Base Address to start storing Input data. The data consumed by the scanner from the target device will be stored starting at this address which can be read by the PLC.
  - Output Size – The number of words (16-bit values) that are produced by the scanner.
  - Output Address – The offset from the PLC's Shared Memory Base Address to start producing Output data. The data to be produced by the scanner to the target device must be written starting at this address.
  - Config Size – The number of configurations words. This value is device specific. Please consult the device vendor for more information regarding configuration data.

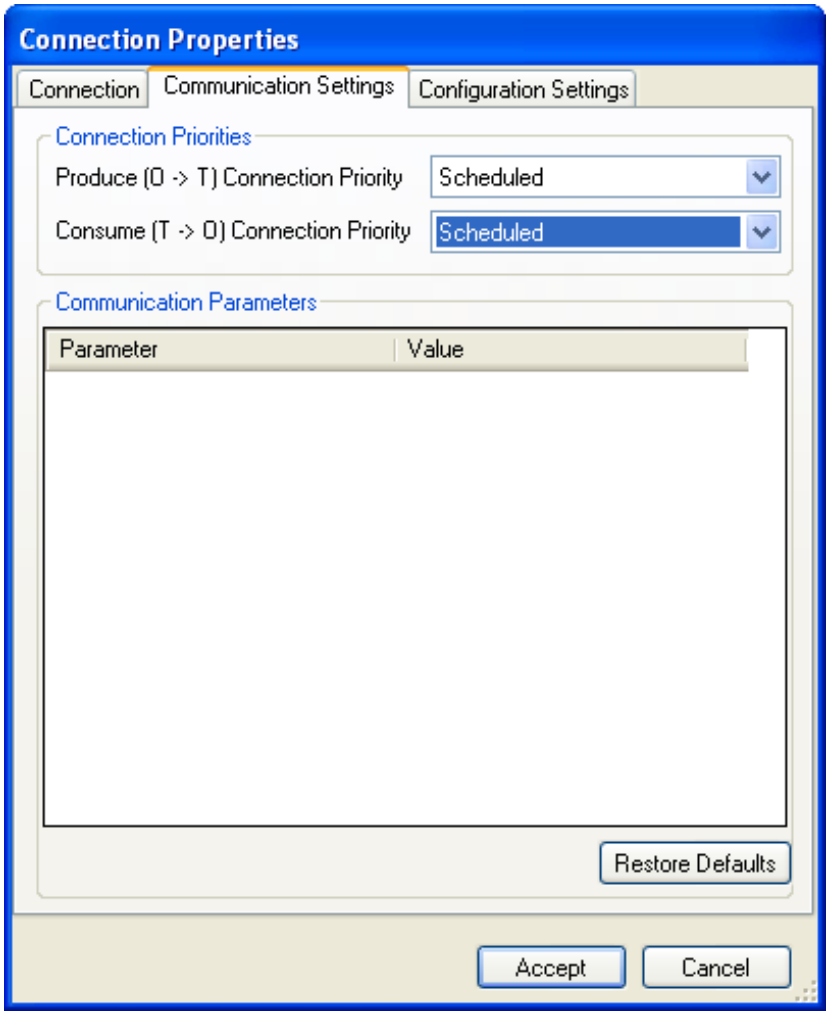


Figure 32: Connection Properties Communication Settings

- Produce Connection Priority – Priority selection for the produce connection. Please consult the server device vendor for more information.
- Consume Connection Priority - Priority selection for the consume connection. Please consult the server device vendor for more information.
- Communication Parameters – List of available communication parameters. The list or existence of communication parameters is device specific. Please consult the server device vendor for more information regarding communication data.

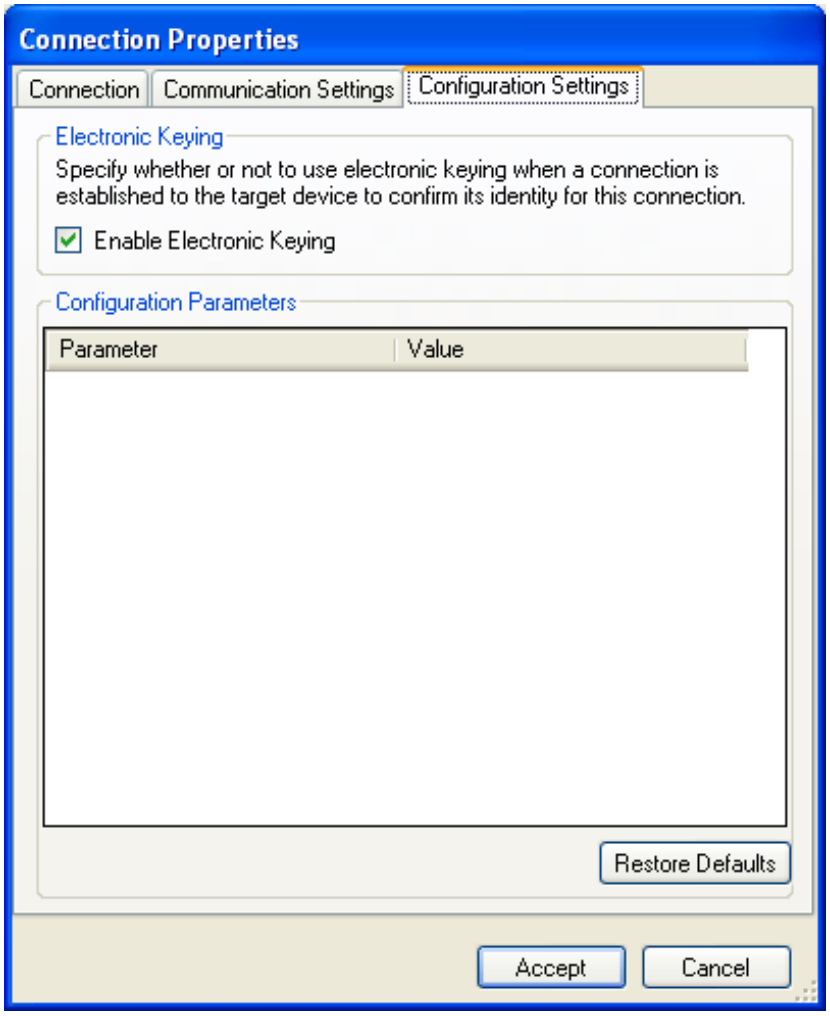


Figure 33: Connection Properties Configuration Settings

- Electronic Keying
  - Enable Electronic Keying – Check to enable electronic keying. Once the Connection Properties are accepted, it will be necessary to ensure that the major and minor revisions in the server device's Device Properties are set correctly. The procedure for viewing and editing the Device Properties for a server device are identical to that for the scanner (see section 10.11).

- Configuration Parameters – List of available configuration parameters. The list or existence of configuration parameters is device specific. Please consult the server device vendor for more information regarding configuration data.

After all Connection Properties have been properly configured, click Accept. See Figure 34.

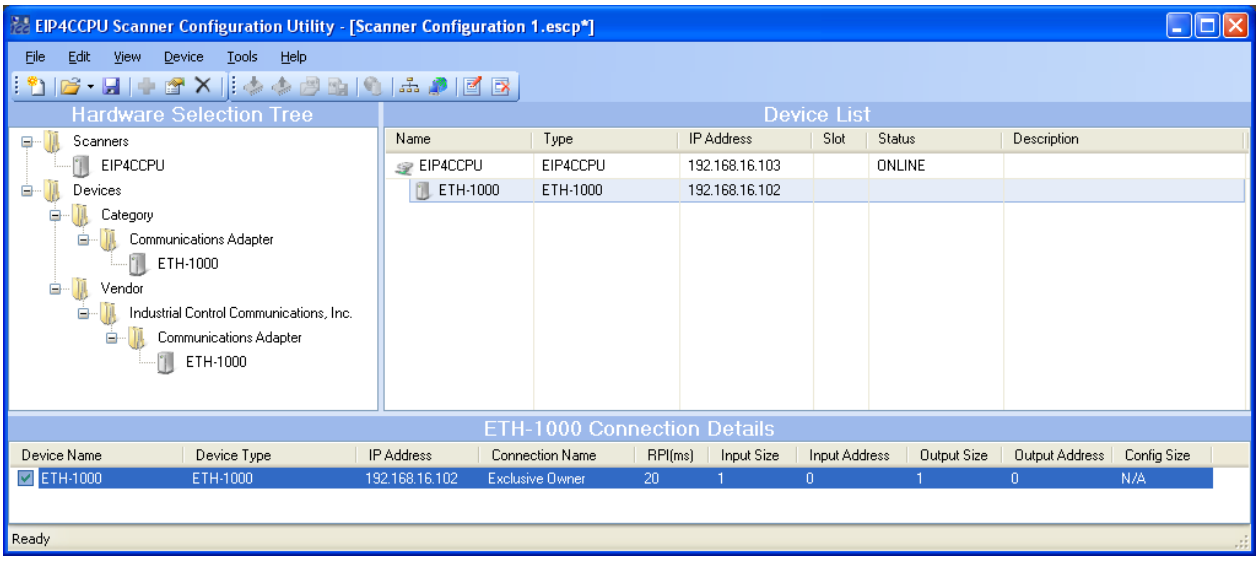


Figure 34: Added a Target and Connection

The target connection will be added to the list of connections in the scanner's Connection Details panel. Selecting the scanner will list all device connections. To list only the connections to a specific device, select that device. The Connection Properties for connections can be edited or deleted.

10.13 Delete or Edit Connections

Connections can be deleted and edited from the Connection Details panel. If all connections for a device are deleted, the device is also deleted from the device list. Execute one of the following methods after selecting the target device in the Device List and selecting the connection in the Connection Details panel:

1. Right click on the connection and select either Delete or Edit Connection.
2. Click on either the Delete or Edit icons in the toolbar.
3. Navigate to Edit->Delete or Edit->Edit.

The connections can also be enabled or disabled by checking/clearing the checkbox to the left of the connection. By default, new connections added to the list are enabled.

10.14 Delete Device

Devices can be deleted from the Device List. If the device contains child devices, those devices will also be automatically deleted. To delete a device, first select the device and complete one of the following methods:

a. Right click on the device and select Delete.
b. Click the Delete icon in the toolbar.
c. Navigate to Edit->Delete

10.15 Downloading the Configuration

Once all connections have been configured for a scanner, the configuration can now be downloaded to the scanner. Before attempting to download the configuration, verify that the Status is not OFFLINE. This will ensure that the utility can establish point-to-point communications with the scanner. The download process will consist of using FTP (File Transfer Protocol) to download the representative configuration file to the scanner's CF card. After selecting the scanner, complete one of the following steps to execute the download process:

1. Right click on the scanner and select Download to Device.
2. Click on the Download Configuration button on the toolbar.
3. Navigate to Device->Download Configuration.

The scanner will have to be rebooted for the configuration to take effect (see Figure 35).

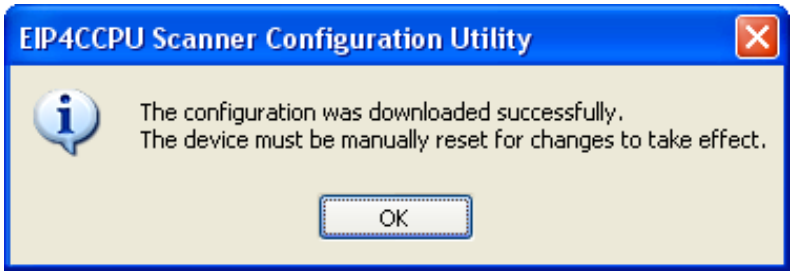


Figure 35: Downloaded the Configuration to the Scanner

10.16 Uploading the Configuration

Once a scanner has been added to the Device List, its configuration can be uploaded and displayed. Before attempting to upload the configuration, verify that the Status is ONLINE. This will ensure point-to-point communications with the scanner. The upload process will consist of using FTP (File Transfer Protocol) to upload the configuration file from the scanner's CF card. After selecting the scanner, execute one of the following procedures to start the upload process:

1. Right click and select Upload from Device.
2. Click on the Upload Configuration icon in the toolbar.
3. Navigate to Device->Upload Configuration.

Click Yes to start the process (see Figure 36).

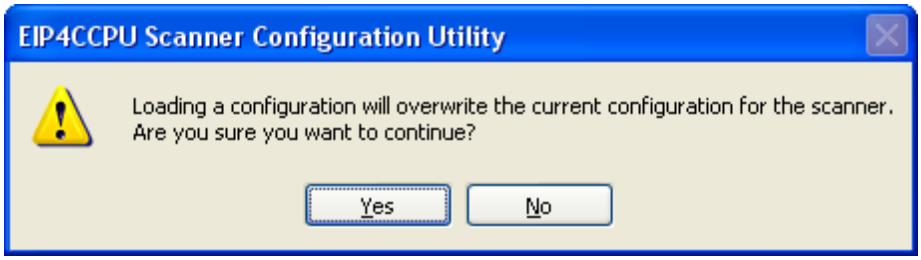


Figure 36: Upload the Scanner's Configuration

10.17 Exporting the Scanner's Configuration

The configuration specific for only one scanner can be saved and exported as an XML file. This allows the user to easily port the configuration to other scanners. After selecting the scanner, complete one of the following methods:

1. Right click and select Export to File.
2. Click on the Export Configuration icon in the toolbar.
3. Navigate to Device->Export Configuration.

Name the file and click Save to save the file.

10.18 Importing the Scanner's Configuration

The configuration for one scanner can be imported and applied to any scanner. This takes advantage of the exported configuration file (see 10.17). Select an existing scanner in the Device List and complete one of the following procedures:

1. Right click and select Import from File.
2. Click on the Import Configuration icon in the toolbar.
3. Navigate to Device->Import Configuration.

Click yes to overwrite the existing configuration (see Figure 36).

10.19 Saving the Project

To retain system-wide configuration and information on all scanners, the project must be saved. Navigate to File->Save As and click Save to save the project.

11. How to Configure Modular I/O

The following is a brief example of how to configure modular I/O. The specific modular I/O used in this example is an Allen Bradley POINT IO 1734-AENTR adapter and 1734-IV8 module. It is assumed that the user has completed installation and setup of the utility.

1. Register the adapter EDS file (1794-AENTR). Refer to section 10.5 for more information on EDS registration.
2. Register the module EDS file (1794-IV8). Each module will have its own EDS file which must be registered before the adapter can reference that module.
3. Once the EDS files have been registered verify that the adapter device has been added to the Hardware Selection Tree (see Figure 37). Note that the module device will not be visible in the Hardware Selection Tree as it cannot directly communicate with the scanner.

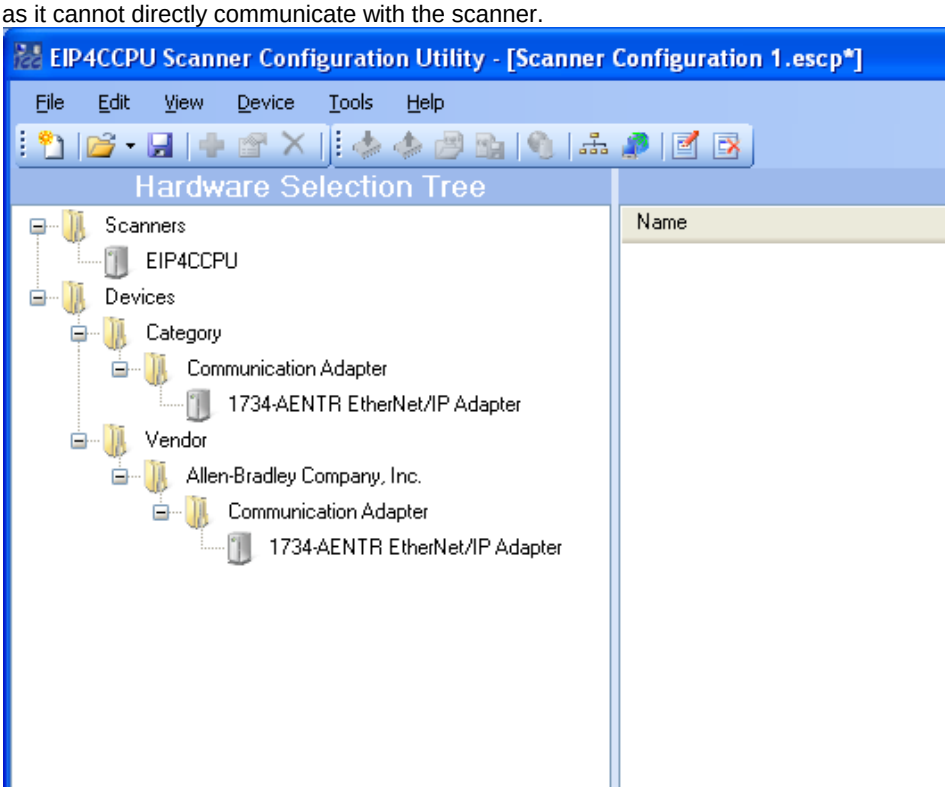


Figure 37: Added POINT IO Adapter to Hardware Selection Tree

4. Add the scanner to the Device List (refer to section 10.9).
5. Add the POINT IO (1734-AENTR EtherNet/IP Adapter) to the scanner. Enter an IP address and click next. If the adapter supports direct connections to itself, select the first type of connection to be created (see Figure 38). Connections to the adapter can always be created at a later time. If the adapter does not support direct connections to itself or the user selects to add a connection to a module, the Module Properties box will be displayed (see Figure 39). All available modules for the POINT IO adapter will be listed.

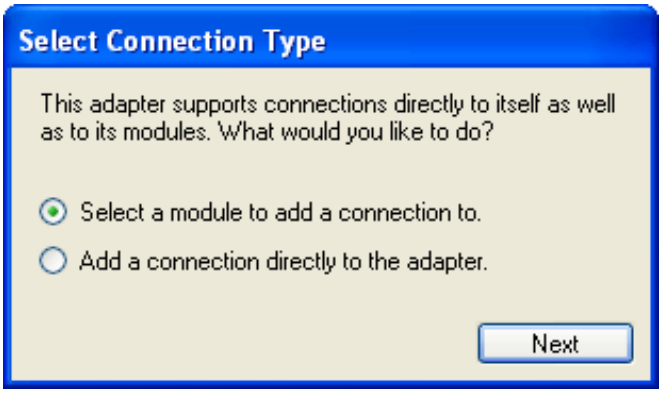


Figure 38: Select Connection Type

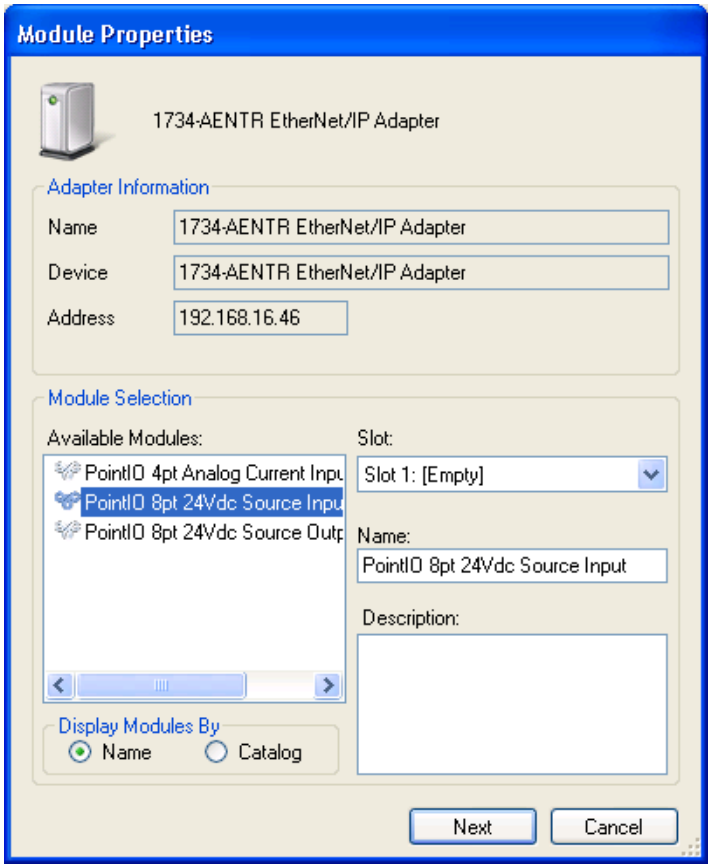


Figure 39: Module Properties

6. Select the 1734-IV8 (PointIO 8pt 24Vdc Source Input) and select the Slot in which the module resides in. Note that the modules can be displayed by Name or Catalog. Click Next to configure the module's Connection Properties (see Figure 40).

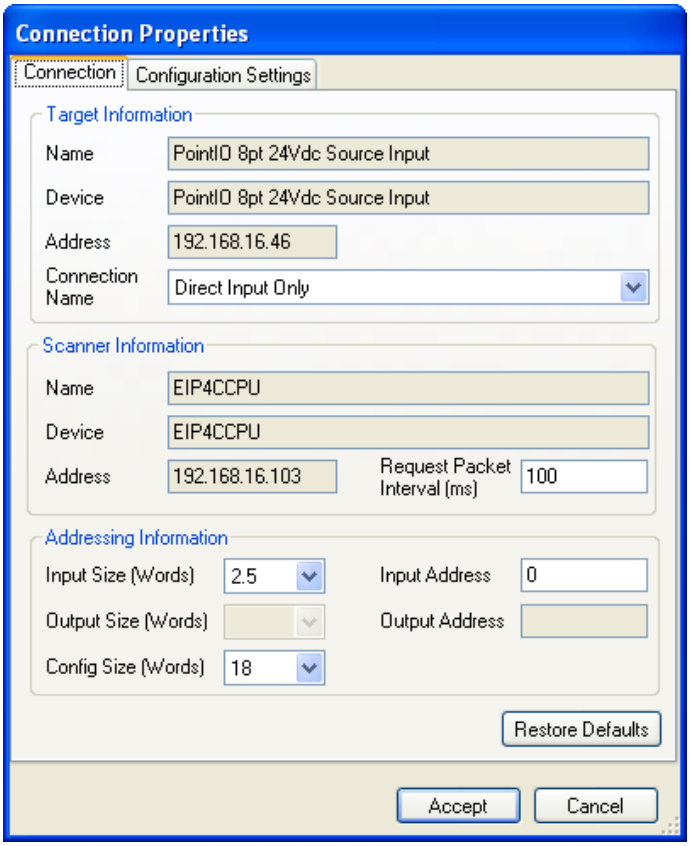


Figure 40: Module Connection Properties

7. Configure the Connection Properties (refer to section 10.12) and click Accept.
8. A modular I/O connection has been successfully configured (see Figure 41).

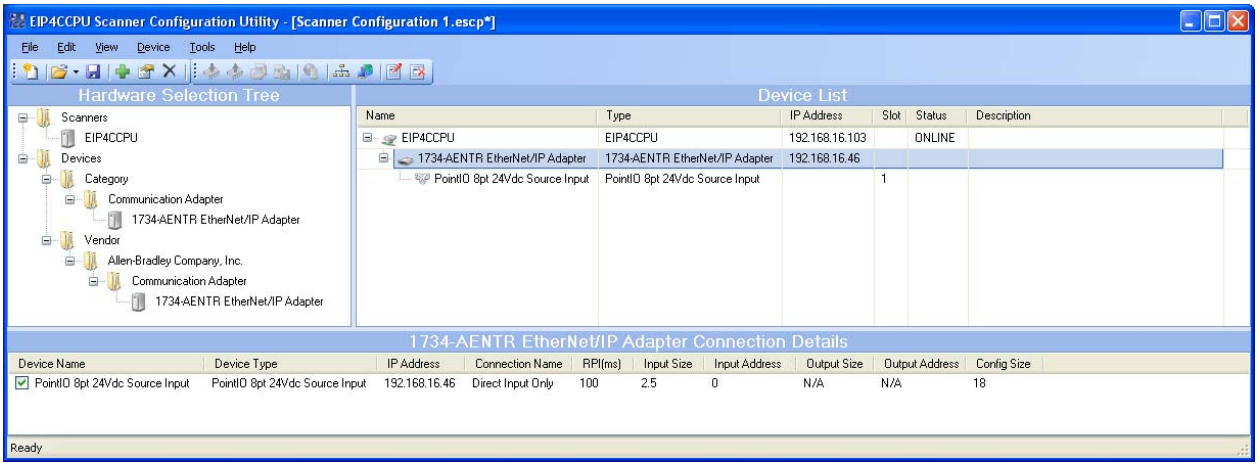


Figure 41: Added a Modular I/O Connection

9. To add another module to the adapter, complete one of the following steps after selecting the adapter in the Device List:
    a. Right click and select Add a New Module to this Device.
    b. Click the Add icon in the toolbar.
    c. Navigate to Edit->Add
10. The process for deleting a module or adapter is similar to deleting a device as described in section 10.14.
11. Now repeat steps 5 to 10 for all desired modules and adapters.
12. Download the configuration to the scanner (refer to section 10.15).

13. Configuration of modular I/O is complete.

12. RSNetWorx Example

The following is a brief example of how to use RSNetWorx to configure class 1 I/O connections between the EIP4CCPU scanner and an FR-A7N-ETH (Mitsubishi F700 drive Ethernet interface card).

1) Configure RSLinx with an EtherNet/IP Driver.

a) Start RSLinx and navigate to Communications->Configure Drivers (see Figure 42).

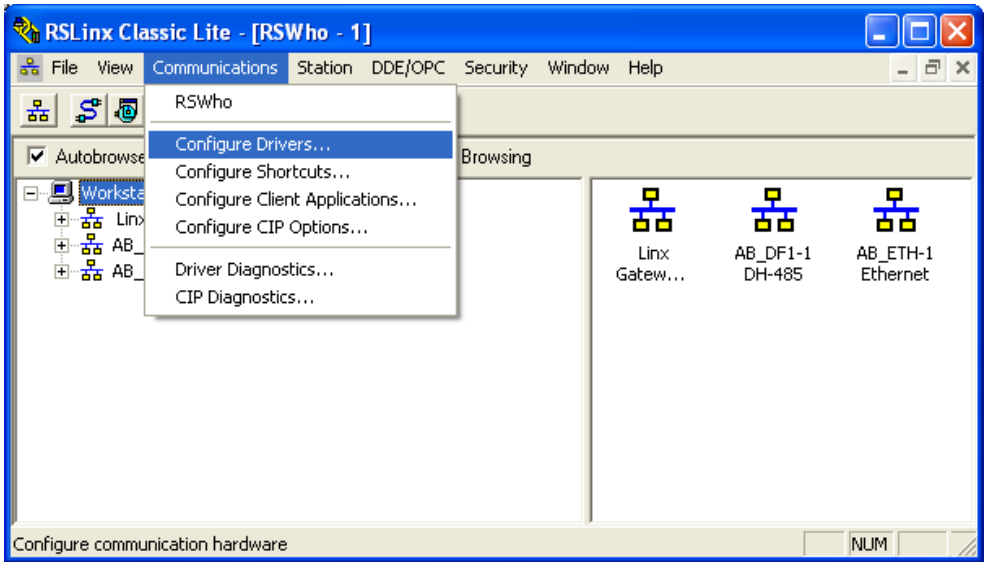


Figure 42: RSLinx Configuring Drivers

b) Select EtherNet/IP Driver and click Add New (see Figure 43).

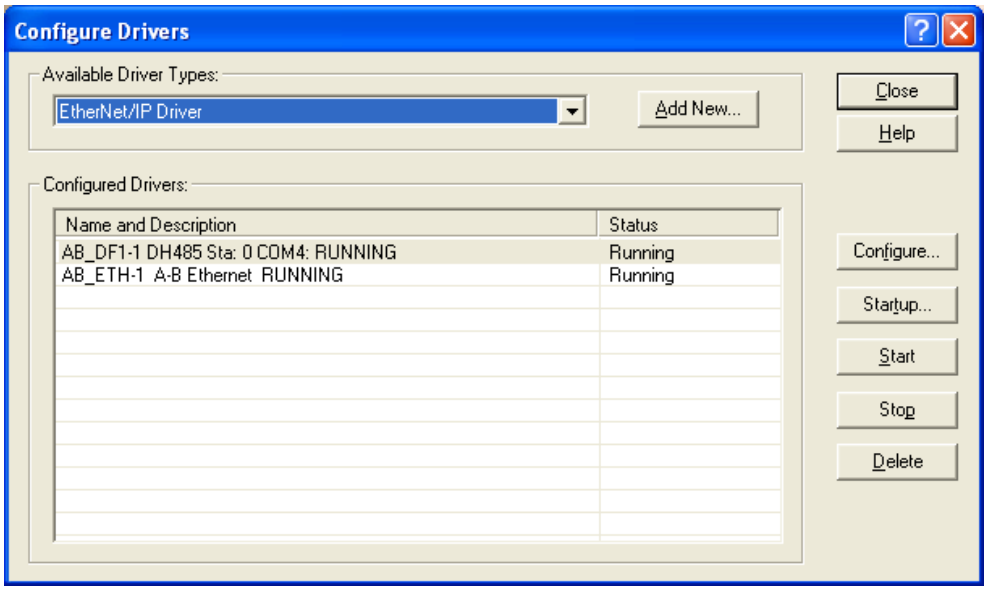


Figure 43: RSLinx Add New EtherNet/IP Driver

c) Select Browse Local Subnet and select the appropriate network interface (see Figure 44).

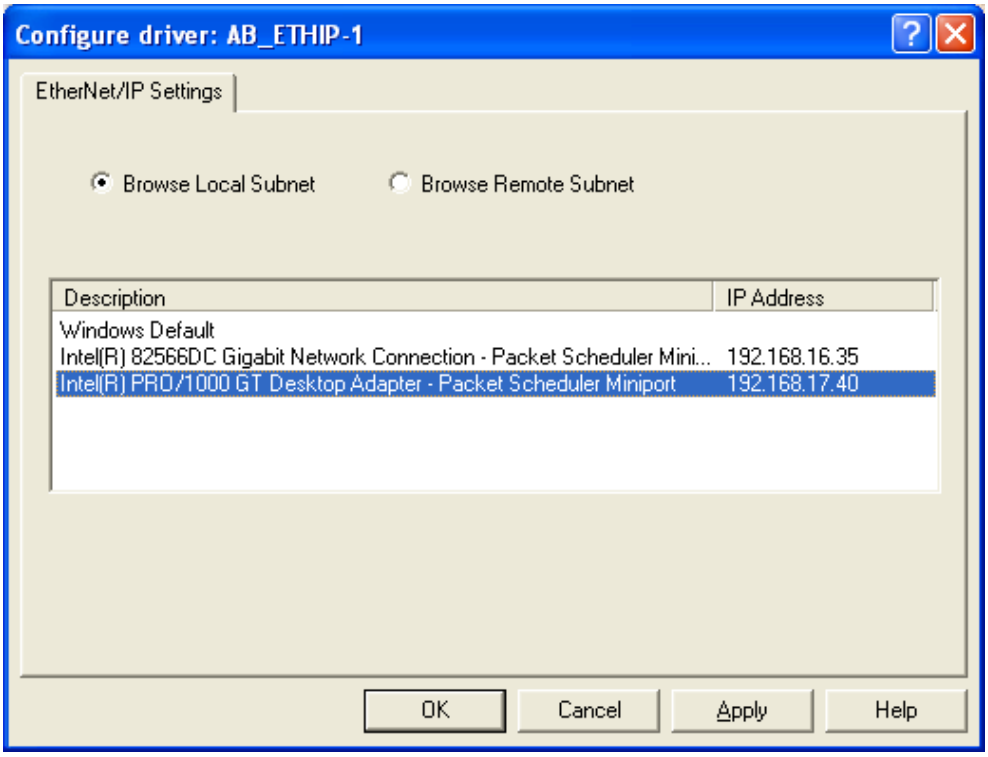


Figure 44: RSLinx Select Browse Local Subnet

2) This completes the RSLinx configuration.

3) Register EDS files with RSNetWorx.

a) Open RSNetWorx and navigate to Tools->EDS Wizard to begin the EDS registration process (see Figure 45).

Figure 45: RSNetWorx Register EDS File

b) Follow the steps and repeat for all necessary EDS files.

4) Set Online Path.

a) Once the EDS files have been registered, navigate to Network->Properties. Click Set Online Path… (see Figure 46).

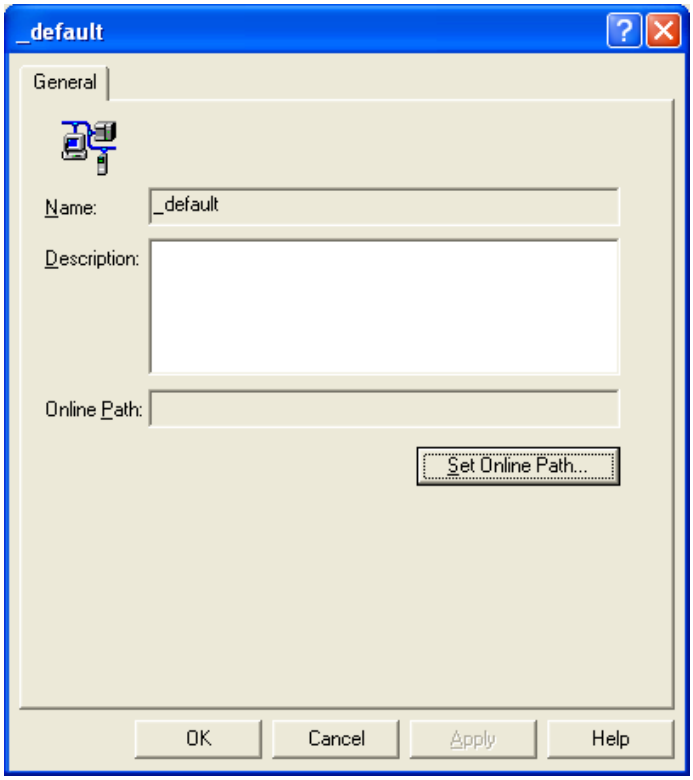


Figure 46: RSNetWorx Set Online Path

b) Choose the EtherNet/IP Driver that was previously created in RSLinx (see Figure 47). Click OK.

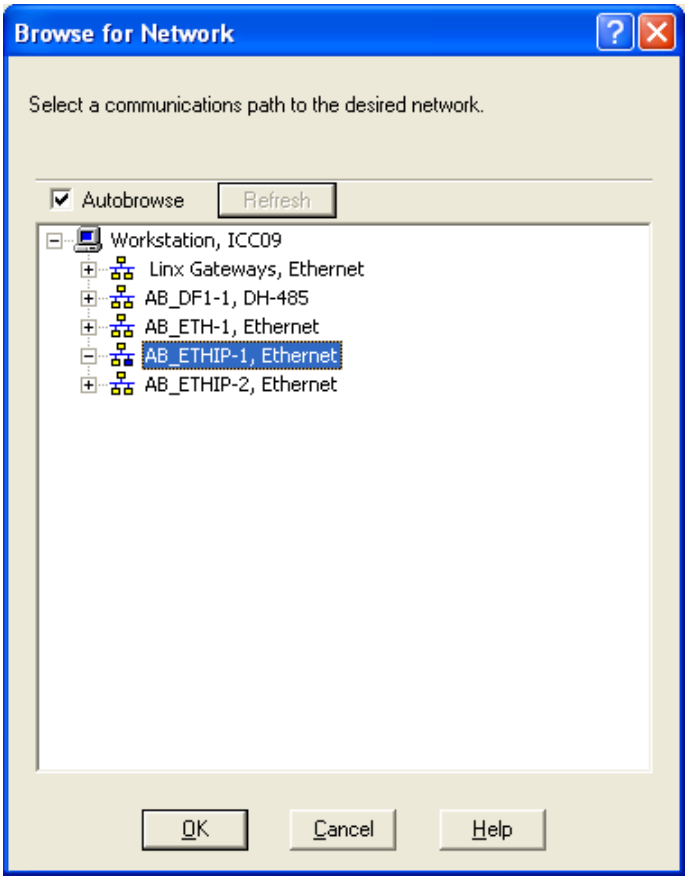


Figure 47: RSNetWorx Select EtherNet/IP Driver

c) Browse the EtherNet/IP network. Navigate to Network->Online. All nodes connected to the network should appear in the graph (see Figure 48).

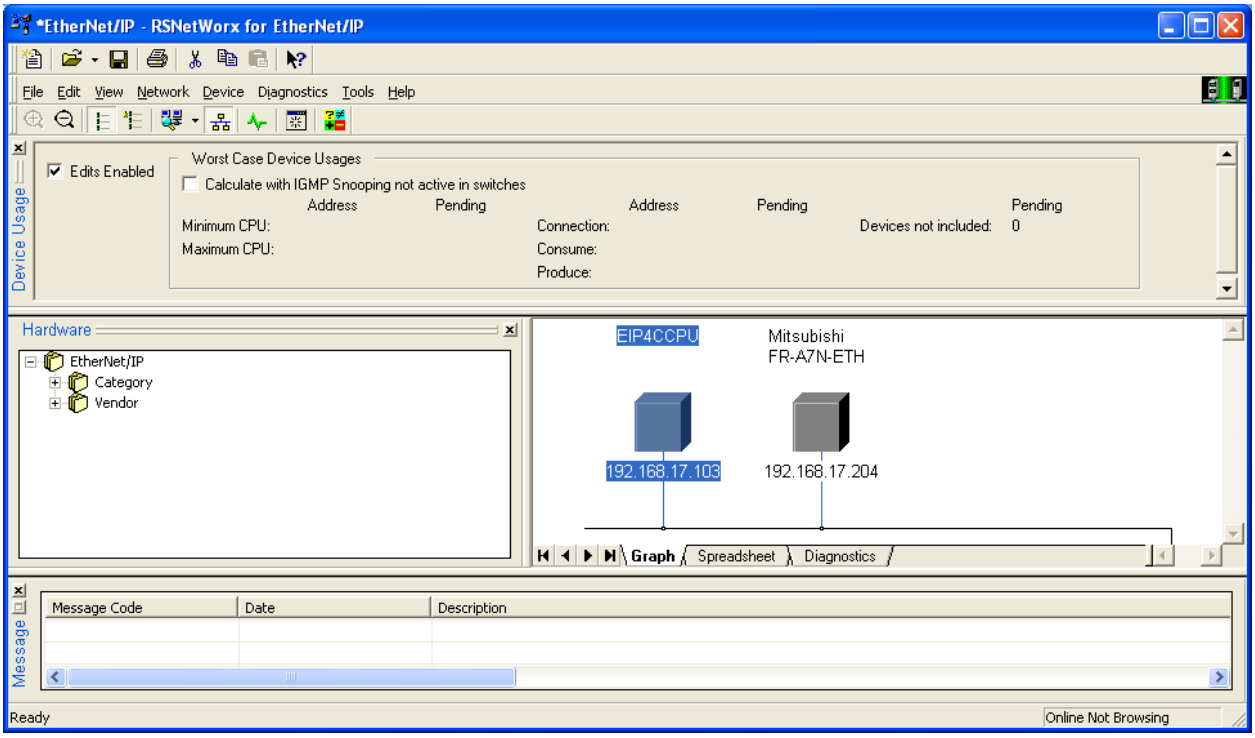


Figure 48: RSNetWorx Online

5) Create scanlist configuration.

a) Select the EIP4CCPU scanner and navigate to Device->Scanlist Configuration (see Figure 49).

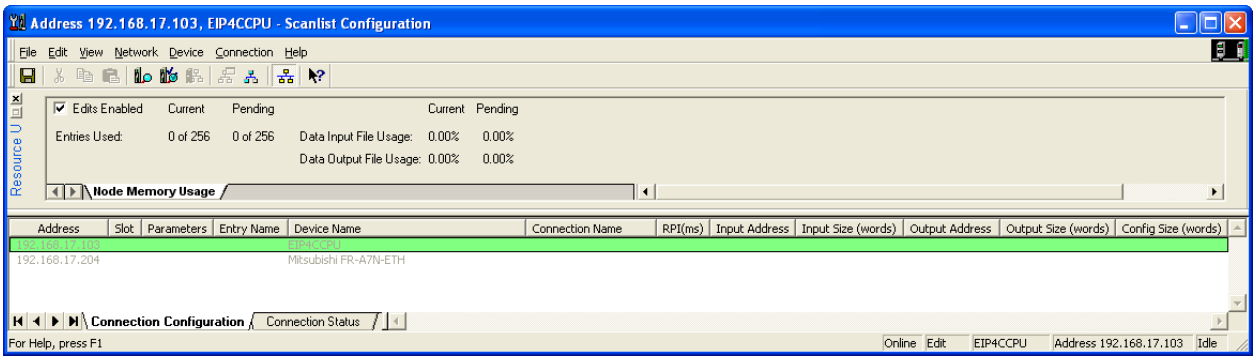


Figure 49: RSNetWorx Scanlist Configuration

b) Select the Mitsubishi FR-A7N-ETH and navigate to Connection->Insert (see Figure 50).

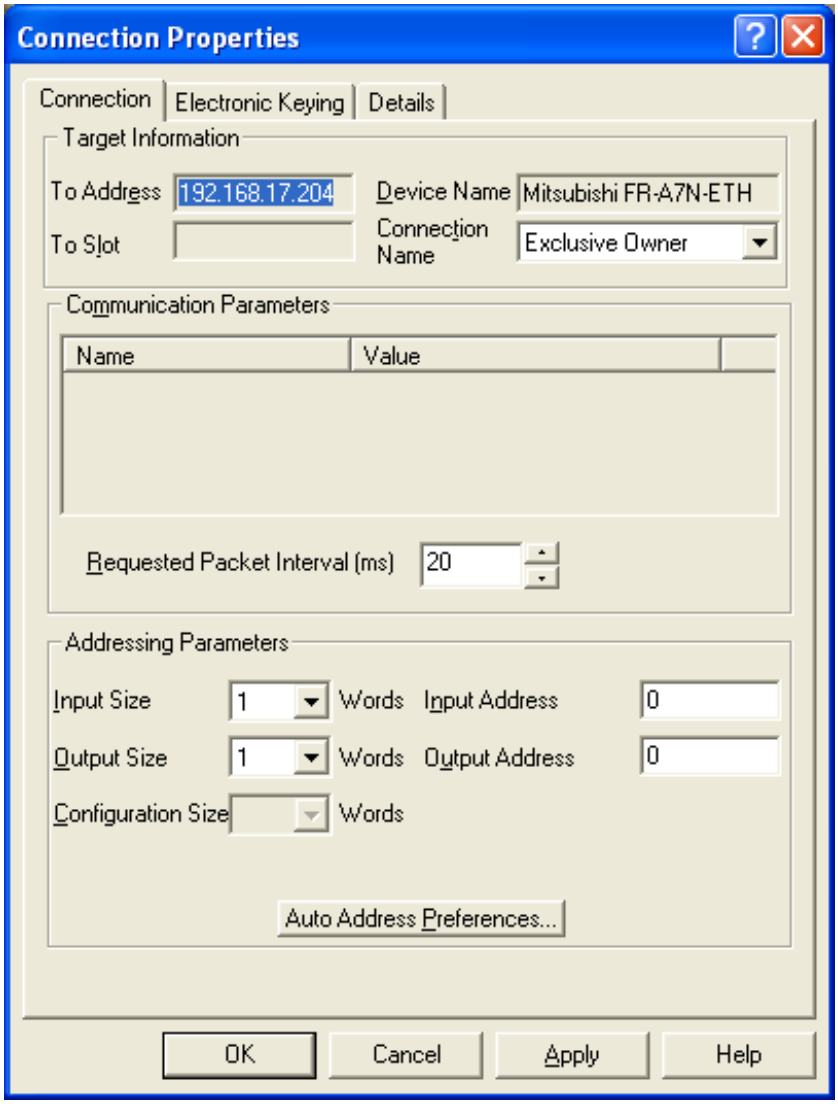


Figure 50: RSNetWorx Connection Properties

c) Click OK to add the connection (see Figure 51).

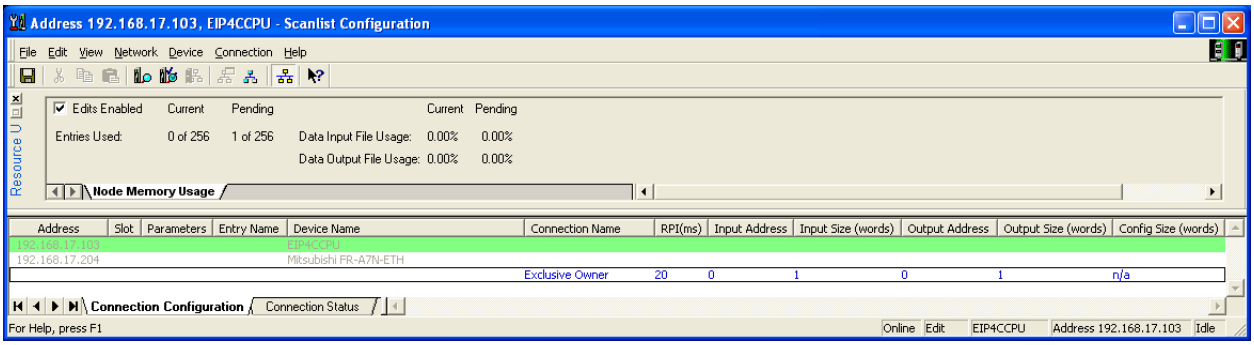


Figure 51: RSNetWorx Inserted a Connection

d) Now download the configuration to the scanner by navigating to Device->Download to Device (see Figure 52). Click Yes to proceed.

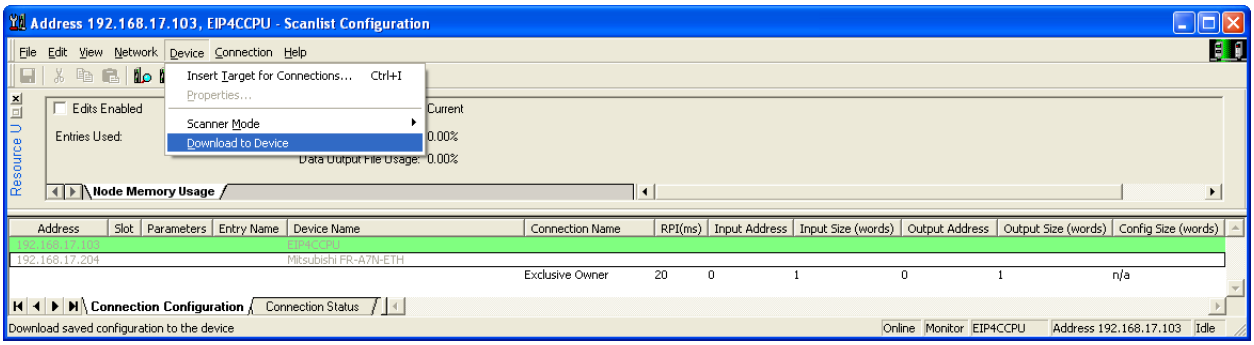


Figure 52: RSNetWorx Download Configuration to Scanner

6) Configuration is complete. The configuration will take effect immediately without rebooting the scanner.

13. FAQ

Question	Answer
How do I obtain a license file?	Refer to 4.1
How do I know if the CCPU is set to run mode?	Refer to 5.1
Why does my scanner show up in the Device Discovery Wizard but not when I click Discover Local Scanners?	The Device Discovery Wizard will discover all EIP4CCPU scanners on the (sub) network regardless of their network settings. Discover Local Scanners will only detect scanners that have compatible network settings. Change the desired scanner's network settings to be compatible.
How can I access the CompactFlash card contents without using a CompactFlash card reader?	The CompactFlash contents can be accessed via FTP. Please ensure that the network settings are properly configured for both the PC and the CCPU before attempting to use FTP.
If the event code is an internal error, what information do I need to provide ICC.	Please email ICC the EIPC_10770A.XML file and the LOG_10770A.XML file. These files are located on the Compact Flash card. Also include your contact information as more information may be requested by ICC.
How do I report problems with the ESCU?	Please email ICC a detailed description of the problem, the steps to recreate the problem, screenshots of the errors, and any applicable EDS files. Also include your contact information as more information may be requested by ICC.
Why can't I configure my device with ESCU?	The EDS file for the device must first be registered with ESCU before the device can be configured. Refer to 10.5.
I have registered the EDS file for an I/O module, but the module is not available to be selected from the adapter.	Only I/O modules registered at the time an adapter was added to a scanner are available in the module selection. If I/O modules for that adapter are added later, a user will have to export (see 10.17) the scanner's configuration and then import (see 10.18) the configuration. This process will update the selectable modules in the adapter's module selection list.
Where is the catalog number?	When adding a module to an adapter, modules can be displayed by name or catalog (refer to Figure 39), After a device has been added to the Device List panel, the catalog identification can be found by right-clicking on the device and selecting Properties. Please note that a device is not required to specify the Catalog identification.
Why is the connection continually timing out (event code 3C)?	The devices may not be able to communicate at the specified RPI due to performance limitations. It is suggested that the RPI be increased or the number of server devices be reduced. Please ensure that there is no extraneous load on the network that may hinder the performance of devices. The use of IGMP-capable switches is recommended to prevent flooding the network with multicast traffic.

Why do I get a socket connection failure (event code 42) when the EIP4CCPU starts up?	The server device may not be online or in a ready state to accept EtherNet/IP connections. Please check the server device for any problems. Additional configuration may be required on the server device. It is recommended that all server devices be powered up before powering on the EIP4CCPU.
Why do I get errors after resetting or power cycling only the CCPU?	If the EIP4CCPU was previously communicating with any EtherNet/IP server devices, resetting or power cycling the CCPU will abruptly terminate the socket and CIP connections. The server devices may transition to an error state and refuse the initial connection from the EIP4CCPU after it starts up. The EIP4CCPU will continually retry connecting with the server devices until it succeeds. It is recommended to review the event history for any abnormal sequence of events before clearing the current error.

ICC

INDUSTRIAL CONTROL COMMUNICATIONS, INC.

1600 Aspen Commons, Suite 210
Middleton, WI USA 53562-4720
Tel: [608] 831-1255 Fax: [608] 831-2045

http://www.iccdesigns.com

Printed in U.S.A